Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 11 Issue 01
2026
Research Article
https://doi.org/10.56979/1101/2026/1422

Scalable Human-Centric Cloud Systems for Efficient Drug Repurposing Leveraging Idle Academic Resources

Omer Irshad[1], Khushbu Khalid Butt[2], Abbas Hussain[3*], Muhammad Faiez[4], Muhammad Majid Hussain[4], and Ali Hussain[4]

[1]Department of Software Engineering, Lahore Garrison University, Lahore, 54000, Pakistan.
[2]Department of Information Technology, Lahore Garrison University, Lahore, 54000, Pakistan.
[3]The College of Arts Science and Technology, University of Management and Technology, Lahore, 54000, Pakistan.
[4]Department of Computer Science & IT , The University of Lahore,Lahore, 54000, Pakistan.
[*]Corresponding Author: Abbas Hussain. Email: abbas.hussain@umt.edu.pk

Received: March 13, 2026 **Accepted:** May 27, 2026

Abstract: In the last few years, cloud-based Information Technology (IT) solutions have become relevant due to providing service-oriented solutions flexibly. Organizations buy cloud computing services as per their needs and pay accordingly. Educational Institutes (EIs) are one of the bodies that consume cloud services to meet their educational, research and development needs. Some of the EIs that face tight budgetary constraints are a bit hesitant to adopt this computation paradigm shift. One of the reasons is that they get concerned about the wastage of the investment that they have already made in their existing IT infrastructure. Consequently, academic research laboratories that require quite extensive computation and storage resources cannot perform their research and development tasks adequately. This study proposes a generic system architecture based on cloud computing. It sets an architectural basis to develop a more adequate resource-sharing computation and storage platform while accommodating the existing IT resources of an EI. For the functional elaboration of the proposed architecture, a methodology dry run is provided. It shows how the proposed architecture enables the better utilization of the existing underutilized resources of the institute. It further elaborates on how these underutilized resources can be made flexibly shareable and accessible to drug repurposing labs. Especially, those labs that extensively and intensively need similar resources to fulfil their computation and storage needs. The statistical analysis shows that the proposed architecture is equally beneficial for both resource-providing and resource-consuming bodies.

Keywords: Cloud architecture; Drug repurposing; Educational systems; Underutilized resources

1. Introduction

Due to the advent of high computing and elastic resource-sharing cloud technologies, the cloud computing paradigm has become relevant in the area of sharing IT resources remotely. It enables the provision of IT resources as a service elastically, flexibly, and on demand. This computing paradigm virtualizes the shared resources and extensively and intensively makes them scalable. It uses the internet, virtualization, grid computing, etc. and provides software, platforms, and infrastructure as a service to its service consumers. Its service-providing mechanism is technically so easy that even consumers with little knowledge of IT can also consume these services. It is very flexible and cost-effective because consumers only pay for what they consume at a particular time. Consumers request resources as per their needs, and cloud service providers charge them and provide the requested resources accordingly. In cloud computing, the resource-providing mechanism is inherently elastic. Consumers can return resources and request more resources from the cloud resource pool if they require them in future. This type of elasticity in resource utilization is obtainable at the infrastructure, platform, and software levels of cloud computing [1].

Currently, many non-IT and low-budget EIs have focused on bringing new IT infrastructure to fulfil their storage and computation needs [2]. Cloud computing is one of the paradigms which best suits such EIs. One of the reasons behind this is, that the cloud provides a service-oriented architecture to its consumers. Low-budget EIs are not capable of buying infrastructure, platforms, and software for each functional unit. Most of the low-budget EIs which are also running biological research laboratories, are suffering relatively more. One potential reason is that biological data is growing heterogeneously at an astronomical rate, and its storage and computation require high-end storage and computation nodes [3-5]. The EIs which cannot afford to buy new hardware and software resources ultimately prefer to build the IT infrastructure with the help of their existing hardware and software resources.

Currently, several architectures for cloud systems are available for developing a cloud to fulfil the needs of different functional units in an EI. The design of most of these architectures is based on the basic components of cloud computing, like provisioning management, identity assurance, security enforcement, resource virtualization, application deployment, charge-back mechanism, etc. [6]. For instance, a privilege to virtualized infrastructure is only granted when it is verifiable, safe, and does not contradict any organization-level policy. These existing architectures do not fully accommodate the existing computational and storage resources of the EI. Furthermore, most of these architectures are domain-specific and work for developing the system for a specific EI. This gap creates a need to develop an architecture that should be generic to implement for all EIs and should also accommodate the existing IT resources of EIs.

Existing resources can be reutilized by making them part of the cloud system. In most cases, the service consumer's end needs not be storage and computation-intensive, so the additional resources can be made available in the shared resource pool for others. This means that all those resources that are being underutilized can be shared with other consumers who require them. Furthermore, several EIs have more or less similar structural requirements for their academic operations [7]. That is, each EI has many functional units that require IT resources for its employees. Based on the structural and requirement similarities of different biological laboratories, common functional components can be identified to develop a generic software architecture.

1.1. Aim of the study

This study aims to facilitate especially those EIs that are undergoing tight budgetary constraints and also running drug repurposing research laboratories that require high computation and storage resources. It further facilitates a community of researchers who require extensive computational and storage resources to perform their research tasks. This study will positively impact society and improve the health standards of life because low budget EIs or even research institutes will start contributing to the research and development domain, which was not possible before or minimal with the existing computing models. To accomplish this aim, a generic human-centric and cloud-based software architecture for fulfilling the storage and computation needs of biological laboratories in the EI is proposed. Its inherent nature is human-centric because it centers around the idea to fulfil the human's elastic and scalable computational needs to uplift the health standards of life. Primarily, it suits well for those biological laboratories which are associated with an EI and running under tight budgetary constraints.

The proposed architecture has two distinct characteristics in comparison with available architectures: (1) the proposed architecture accommodates the existing resources of an EI (i.e., computational and storage resources, network, etc.). All the resources become part of the cloud and flexibly share available storage and computational capabilities with other consumers who need them. Due to this, additional resources from other functional units of the EI can be provided to its biological laboratories (2), As the proposed architecture is developed on the common needs and structural similarities among different EIs, it is generic for all EIs for system requirements, design, and implementation.

The proposed architecture logically divides the whole EI into different functional units. Each functional unit is further divided into different service-consuming bodies like faculties, laboratories, libraries, etc. Each consumer can be divided (if required) into sub-consuming bodies. Resources are granted to functional units, and each functional unit further divides the given resources among its sub-consuming bodies as per their needs. Due to such similarity in requirements and structural hierarchy among most of the EIs, any EI can use this architecture with slight modifications [7].

1.2. Research Work Organization

This research is organized into the following sections. Section 2 provides related work on existing architectures for developing cloud-based solutions in the EIs. The related work is logically divided into three parts: (1) it presents architectural aspects of cloud computing, including the characteristics, services providing, metering, and deployment aspects, etc., and (2) it explains the existing cloud-based architectures for EIs. Furthermore, it highlights some commonalities and differences held between existing educational cloud architectures (3) and elaborates on some potential problems in existing educational cloud architectures. Section 3 describes the proposed solution. Section 4 elaborates on the proposed solution with the help of a methodology dry run. Section 5 concludes our research contributions and provides some future direction.

2. Related Work

Cloud computing is an emerging and evolving paradigm that has been gaining rapid focus for the past few years. It encompasses reliable, elastic, cost-effective, highly accessible, dynamic, and user-friendly IT solutions for the industrial and educational sectors. It relies on developing the infrastructure once to use it multiple times in shared and multiple ways. The NIST defines cloud computing as, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1, 2].

2.1. Cloud computing service models

Cloud computing mainly focuses on service-oriented and deployment-oriented models. The service model includes demand-based, customized, rapidly elastic, self-managed, globally accessible, resource-pooled, and metered service. Specifically, the demand-based and self-managed characteristics emphasize the fact that cloud computing services can be readily accessible or demanded as per one's custom needs and without the intensive involvement of the provider of the cloud services. Global access to the services is made possible with the help of clearly defined and widely acceptable service access standards. These service access standards cop to minimize the issues related to cross-platform heterogeneity and uniformity of globally dispersed resources Due to this, cloud services can be transparently accessible through high-end machines or handheld devices like PCs, cell phones, smart watches, etc. Due to the standard and well-managed mechanism of resource integration and their pooling, the service consumer can seamlessly request new resources or modify the existing resource acquisition plan as per the current need. So, this fact leads a service consumer and provider to dynamic, monitored, controlled, measured, and customized metering for the service cost.

Moreover, the basic models for cloud services facilitate the service consumers at the software, application, platform, and infrastructure levels. Software-level services provide various applications developed and managed by third-party developers. Consumers of the software services can access these applications through the internet and web browsers and directly run them without first downloading and installing them on their smart devices. Platform-level cloud services facilitate users to create custom platforms based on various cloud frameworks to develop, deploy, and manage customized applications. Cloud platforms are accessible over the web and enable developers to concentrate on the development of computer applications instead of spending time and effort on managing the system software, software updates, firewall-based protection, storage, computation power, etc.

Infrastructure-level cloud services facilitate users, especially with hardware components like storage, computing, networks, and servers, etc. The consumers of these cloud services focus on managing the functional and management aspects of the computer application instead of dealing with resource installation and configuration-level issues. The infrastructure-level service provider takes the responsibility to provide such a configurable environment in which all the system and application software can perform correctly and accordingly to meet the needs of the service consumer.

Based on the budgetary, accessibility, and privacy constraints of the organizations, four types of models are fundamentally available for building cloud-based solutions. (1) The private cloud is developed for a specific organization. The organizations internally manage the cloud and provide access to services to their consumers through the organization's intranet. The infrastructure, platform, and data are made private and secure by the firewalls and other encrypted communication mechanisms. (2) The public cloud shares IT resources among the subscribers of the cloud services. All the infrastructure-level resources are deployed and managed by the cloud service provider instead of the cloud service subscribers. The cost-benefit model is mainly based on the advertisement and those subscriptions that require resources beyond the limits of free access. The services of this model are usually preferable to solve complex problems that require advanced and large-scale availability of computation and storage resources. (3) The hybrid cloud is a blend of a private cloud and a public cloud. Organizations maintain a private cloud for their sensitive data and operations and opt for public cloud services for public or non-sensitive data and operations to reduce costs. (4) The community cloud is the latest variant of the private cloud. It is developed for a specific community of a business type that requires similar business flows, standards of compliance, software, platforms, privacy, and security needs.

Currently, a lot of educational institutions that are running biological research laboratories are facing tight budgetary constraints, especially for building IT infrastructure. The need for building IT infrastructure is triggered by the requirement for extensive storage and computational power required for performing biological research and development tasks. For example, in 2024, the Nucleic Acids Research journal reported 1959 biological databases that are publicly available [8]. The report shows that, in 2024, 97 new databases were reported and 388 databases were eliminated in comparison to the report published in 2023 [9]. Furthermore, some primary databases like GenBank [10], DNA Data Bank of Japan [11], European Nucleotide Archive [12], etc., reported an increase of petabytes in their data in the last few years. In addition to the exponential growth of biological data, several biological systems are publicly available for the processing of such huge and multidimensional data [13 - 28]. Some such systems require a huge amount of storage and computation resources, especially when they are used for offline data processing. For example, in the case of warehouse-based systems, both a large amount of data and computation resources are required at the same time [29].

To cope with the issues related to the management and processing of data that is astronomically growing in multiple types and formats, cloud-based solutions are relevant for their scalability, elasticity, and cost-effectiveness [30, 31]. Furthermore, a significant part of these solutions is based on existing technologies like the internet, virtual platforms, etc., that make them user-friendly and easy to opt for. Their need-based resource elasticity and pay-as-you-go characteristics make them distinct from the existing solutions in this domain. Specifically, several cloud-based system architectures have been proposed in the last few years in the domain of education systems that have research laboratories as well. In [32], a proposed architecture based on centralized resource sharing is proposed. According to this architecture, there is a centralized cloud that connects multiple geographically dispersed sites of institutes. Each site has multiple systems that are privileged to demand excessive resources. There is a local site manager at each site who collects the resource access requests for all the systems and sends these requests to the centralized cloud in a summarized manner. Similarly, the centralized cloud receives resource requests from other sites and, as per the policy, grants access to the resources to each respective site.

2.2. Cloud-based system architectures

Customization and cost-effectiveness are the prime requirements when a cloud-based solution is required to be deployed. A model in [33] meets the academic and administrative needs of the staff while keeping customization in place. This model facilitates its end uses by rapidly developing a customized application instead of managing resources at the infrastructure and platform levels. In [2], a shared and centralized cloud architecture for a community of education systems is proposed. Multiple EIs can be connected to the centralized cloud as per the agreed service level agreement between the service consumer and provider. The prominent characteristic of this cloud is its extensibility to accommodate new EIs. Moreover, it is also adaptive to accept upcoming changes either at the service level or structure level for connected EIs.

Some cloud architectures opt for the layered approach to model a cloud computation system. They divide the whole system into multiple layers of functional components of the system. These layers exchange information with each other and to its end users through a proper service-sharing mechanism.

For instance, in [7], an architecture divides the whole system into three layers. The first layer serves the requests of end users. The second layer manages all the software, platform and infrastructure-related issues. This facilitates the end users to focus on the actual task to be performed instead of messing up with the network, software, hardware, configuration, installation, licensing, etc., related issues. The third layer manages the deployment-related tasks, like which cloud deployment model (i.e., public, private, community, hybrid) is more appropriate than others, by keeping the needs of the end user, how the deployment will align well with the rest of the two layers, what are the deployment constraints or limitations coming from the underlying domain, etc? These layers are loosely coupled with each other and so can be modified for the different universities. For example, at the third layer, if a university needs a private cloud instead of a community cloud, then it can easily be changed by keeping the rest of the two layers unchanged.

Cloud computing is also significantly easing the accessibility and scalability of electronic learning systems [34]. The scale of creating and sharing educational content is exponentially increasing cost-effectively. Content providers and content receivers can exchange information online and are no longer required to be geographically in the same place and time. The increasing need for networks, software, platforms, and infrastructure is no longer a limitation for content creation, management, and broadcasting. The cloud services for this domain are now at the doorstep of all the service-consuming bodies in a measured and flexible manner [35].

The cloud architectures for electronic learning systems that are component-oriented divide the systems into multiple structural components. In [36], an architecture for an electronic learning system divides the whole system into five structural components. First, the infrastructure component elastically manages the hardware resources and ensures timely availability. Second, the software component encompasses all the application software that is required for application development. Third, the middleware component manages the system software, like the operating system and other resource management utility software. Fourth, the application component integrates, manages, and shares all the electronic learning resources to its consumers, like digital libraries, online data centers, learning content, etc. Fifth, the service manager component manages the cloud services for its consumers at the infrastructure, software, platform, and application levels.

Private clouds are also being deployed for distance learning models. In this, multiple systems like online learning and communication, learning content administration, virtualized laboratories, etc., are preferably deployed in the community or specifically on the private clouds [37]. All the administrative tasks for the instructors, students, and other staff members are performed on these cloud-based systems in a customized and personalized manner. In comparison with other offline systems, cloud-based systems are more relevant due to their accessibility, availability, and scalability for learning. These systems understand the teaching and learning abilities of their instructors and students, respectively, and accordingly suggest better and customized teaching and learning paths for easy and quick understanding. Such educational systems encompass the online delivery of lectures, helping material in soft form, audio and video-based streaming, assignments, quizzes, etc., for content delivery to students and their periodic evaluations. The virtualized environment for facilitating students with online laboratories helps to elastically grant hardware and software resources to individuals or groups as per their requirements without the interaction of laboratory attendants. Furthermore, students and instructors are free to be geographically anywhere to communicate with each other, even with their smartphones or handheld devices, to complete the assigned task.

The prime focus of the majority of the proposed EI architectures is to improve the accessibility and service quality of the system services while significantly reducing the overall system cost. These architectures exhibit centralized control over the system resources and provide their accessibility to the end user based on a pay-as-you-go model [33]. The structural components of these architectures are based on the virtualization of hardware and software, and service provisioning, costing, configuring, metering, policing, etc. On top of these, ensuring privacy, security, and identification for both service provider and consumer exhibit umbrella structural components at both virtualized hardware and software levels. Moreover, these architectures exhibit common characteristics for pooling and sharing software and hardware resources elastically, educational content, applications for quick learning, timely availability and accessibility of services globally, etc.

2.3. Research contributions

So, the intensive and extensive analysis provided in this section shows that several architectural features should be included to make an architecture usable for those low-budget EIs that require cloud-based solutions. Existing architectures are lacking or partially provide support for these architectural features. For example, (1) the majority of the architectures are domain-specific, like EI. (2) They do not make the existing resources of EI a part of the cloud. Existing resources will get wasted if any EI selects from available architectures (3) They do not provide a sufficient mechanism for how the services will be charged back, provisioned and measured if the service-consuming end also contributes its resources to the resource pool of the cloud. Table 1 shows the architectural features that are available and should be in existing architectures.

Table 1. Architectural features that are required to be in existing cloud architectures.

Architectural features	Existing cloud architectures	Need to be in the existing cloud architecture
Virtualized infrastructure and applications	Yes	Yes
Provisioning mechanism	Yes	Yes
Metering mechanism	Yes	Yes
Charge-back mechanism	Yes	Yes
Policy management	Yes	Yes
Configuration management	Yes	Yes
Security and identity management	Yes	Yes
Development	Yes	Yes
Accommodating existing Hardware resources	Partial	Full
Generic for domain	No	Yes

Based on the architectural features provided in Table 1, a comprehensive architecture for low-budget EIs is proposed. The proposed architecture is generic for all such EIs and capable enough to accommodate the existing resources while ensuring a reliable, elastic, and maintainable cloud-based solution.

3. Methodology

In general, the EIs have several structural similarities for efficiently running their educational and operational matters. They have some faculties, libraries, laboratories, research centers, administration, finance, and procurement departments, etc. Each department may have its intra-departmental administration office, digital library, laboratories, etc. This means a department may further be structurally divided into sub-units horizontally or vertically for better administration and workload division. The need for IT resources in each department can categorically be identified for all its sub-units based on the roles and responsibilities. For instance, a computer science department may require computer systems for its students, faculty members and administration staff, computation and storage-intensive servers for student labs and research labs, software and internet for faculty members, students, and other staff members, etc. Additionally, all the stakeholders who use IT resources in a department may have different needs at different times. Figure 1 hierarchically represents the structure of an EI, including its department/functional units that demand the IT resources.

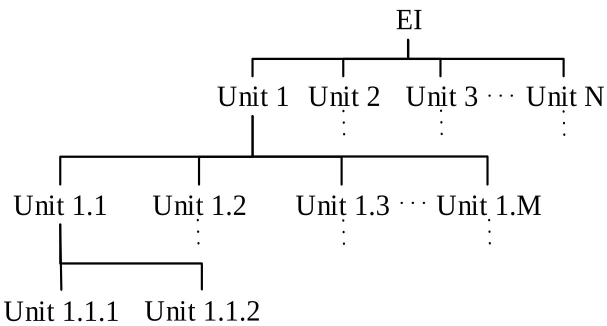

Figure 1. Hierarchical representation of an EI for its resource-consuming structural constituents

3.1. Cloud Unit

All the tree nodes that are shown in Figure 1 represent functional Units of EI. Each Unit can have its child units called Sub-Units. The leaf nodes represent those smallest Units that cannot be further divided into Sub-Units. For instance, an EI can have multiple faculties as Units (i.e., Unit 1 to Unit N.), and each faculty has its Sub Unit (i.e. Unit 1.1.), like the administration department that is further divided into two non-divisible sub-departments (i.e. Unit 1.1.1 and Unit 1.1.2.), like staff administration and student administration. Moreover, the depth and breadth of this hierarchical structure define the scope of the cloud system for its components and the functional association among them.

The hardware and software needs of each leaf node can be different and variable based on the time and nature of the work to be done. One observed fact is that the resources assigned to each Unit are not always in use at their full capacity. This means most of the Units waste the assigned resources in the time when they are not using them or partially using them. These free resources can be added to the resource pool of EI Private Cloud so that they can be used by another Unit. The following equation shows the amount of free resources that can be contributed to EI Private Cloud for a date and time.

Free Resources = Assigned Resources – Resources Being Used

The above equation contains three variables. The Free Resources variable shows at a particular time how many resources are free and can be contributed to the EI Private Cloud. The Assigned Resources variable contains the amount of resources assigned to a Unit at a particular time. The Resources Being Used variable shows the amount of resources that are being used at a particular time. This equation helps to keep track of resource needs or possible contributions by each Unit at a particular time. The Units that share their free resources with others via EI Private Cloud are called Contributors. Based on the contribution, the EI Private Cloud returns other resources to its Contributor as per the resource exchange mechanism. For instance, if a Unit shares 2GHz computational power for some time then EI Private Cloud will compute its worth and, accordingly in return, grant access to the deployment server if it qualifies. In the case of disqualification, the other resource exchange deals will be offered, or the contribution will be added to the Contributor's account for later use.

3.2. Cloud Administrator

To implement the resource exchange mechanism, each unit has a local server that manages the rest of the systems in that Unit. The local server acts as a coordinator that connects all the systems in the Unit with the EI private Cloud through the Administrator (i.e., a computational system). Figure 2 shows the basic components of the Unit and how it connects systems with the local server.

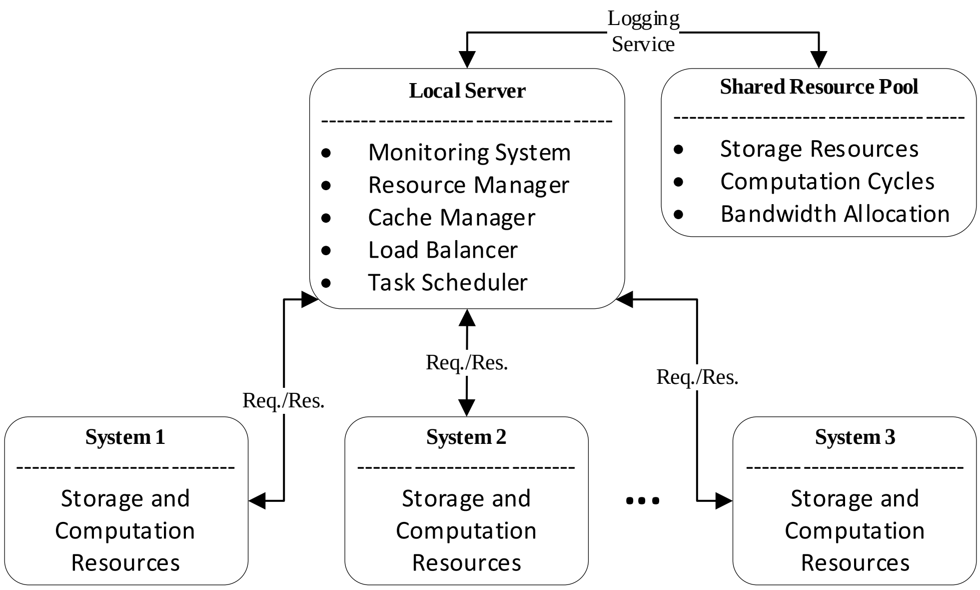

Figure 2. Architecture of a Unit

Figure 2 shows that all the systems in the Units are connected with the Local Server and bi-directionally communicate with it. The Local Server maintains the contribution log for each system that contributes its free resources to the EI Private Cloud. The Local Server receives the free resources from systems and communicates the resource availability information to the Administrator so that these resources can be added to the resource pool of EI Private Cloud. Similarly, if a system requires additional resources from the EI Private Cloud, it sends a resource request to the Local Server, and the Local Server performs a couple of actions to fulfil the request. First, the Local Server evaluates whether the requested resources can be granted locally by the Unit or not. If the case is no, the Local Server takes a second action and requests the EI Private Cloud through the Administrator for the requested resources. In both cases, the Local Server logs and acknowledges the statistics of the requested and granted resources to the EI Private Cloud through the Administrator so that periodic statistical reports can be generated and analyzed to make resource utilization more efficient in future. The statistical report encompasses parameterized and quantitative information like, which system of the Unit contributed what amount and type of resources, what was the resources contribution time, which Unit contributed the maximum and minimum amount of resources, what are the idle time of the systems, whether the resource request by a system was fulfilled locally or by the EI Private Cloud, how the Unit was paid-back or charged by the EI Private Cloud, etc. All such types of information logging and its analysis are done by an efficient coordination of the Unit's Local Server and the EI Private Cloud through the Administrator. Figure 3 pictorially depicts the role of the Administrator and the request and response mechanism between all the Units and the EI Private Cloud.

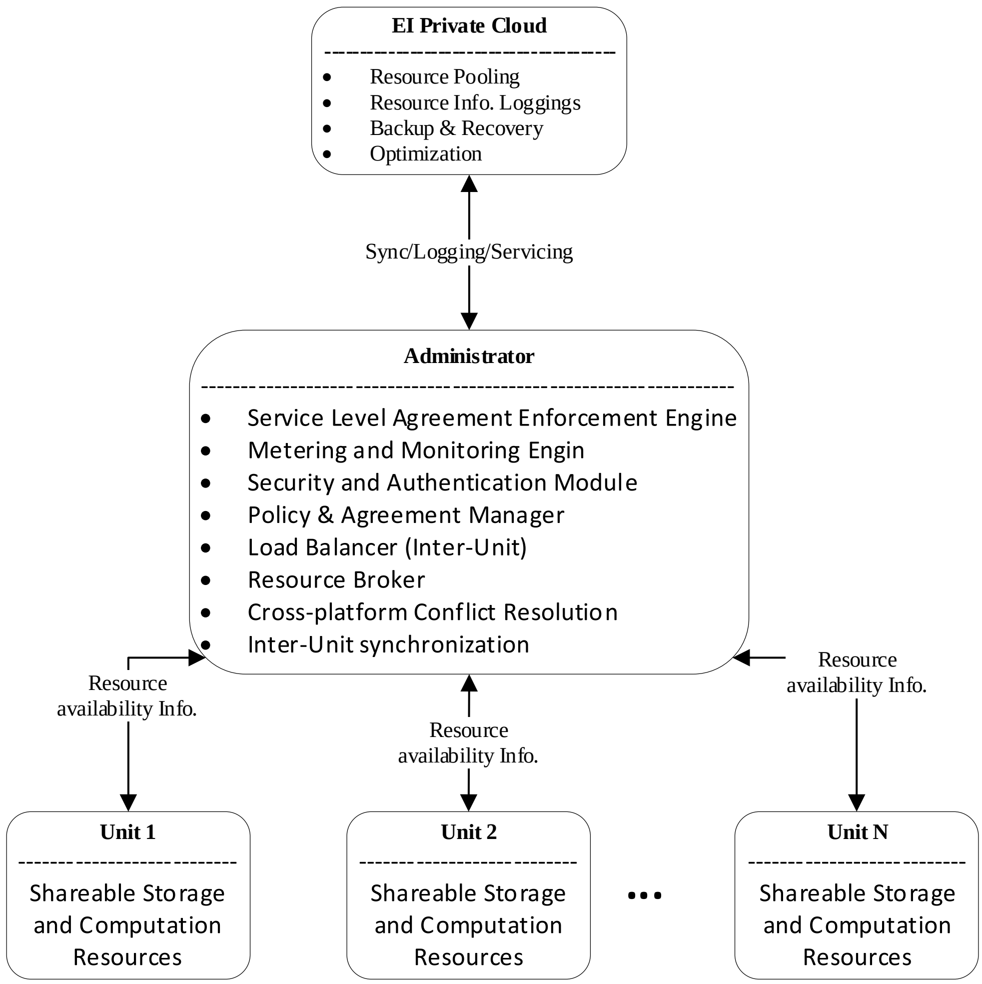

Figure 3. Architecture of the Administrator

Figure 3 shows that all the Units are connected with the EI Private Cloud through the Administrator. The Administrator receives the resource-sharing offer from the Local Server of the Unit. In case of non-compliance to the agreed resource-sharing policy, the Administrator will reject the resource-sharing offer and will send the resource-sharing policy to be compliant with. If the Administrator accepts the offer, then the Administrator will add the resources to the resource pool and will also notify the EI Private Cloud. The following are some of the major responsibilities of the Administrator in the context of establishing coordination between Units and the EI Private Cloud.

- Calculate and add the received resources to the resource pool of the EI Private Cloud and send an intimation to EI Private Cloud.
- Estimate the aggregated and categorical resource contribution by the Unit for reporting purposes.
- Log the resource contribution in the Administrator log file for pay-back and efficient tracking in future.
- Ensure compliance with the predefined resource exchange agreement between all the Units and EI Private Cloud.
- Maintain all the information regarding the resources granted to each Unit on its request.
- Provide a transparent interface to the Unit if resources of EI Private Cloud are being used by the Unit.

So, the Administrator is a significant functional component of the whole system. All the matters regarding the resource request approval, grant, denial, log, compliance, pay-back and charge-back estimation, etc., are settled by this component.

In addition to the functionalities of the Administrator, the EI Private Cloud is the backbone of the entire cloud computing system. Although the Local Server of the Unit also locally fulfils the resource requests of its systems, it does not provide services as EI Private Cloud does. The EI Private Cloud provides services at infrastructure, platform, and software levels to all its Units and their connected systems. Figure 4 pictorially demonstrates the layered architecture of the EI Private Cloud encompassing its basic architectural components.

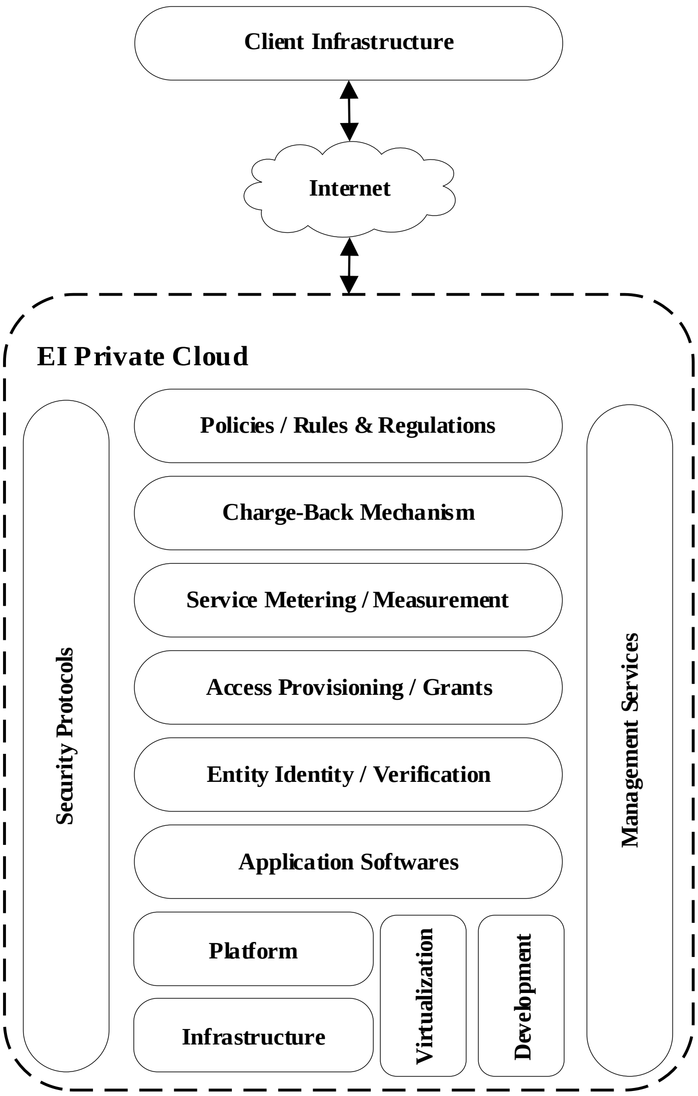

Figure 4. The architecture of the EI Private Cloud.

Figure 4 shows that the individual layers that manage the policing, charge-back, metering, provisioning, identification, and security mechanisms are based on the development and virtualization of infrastructure and platform. Additionally, all these layers collectively facilitate the software management layer for building an environment to develop, deploy, and manage software applications. These layers are discussed in the subsequent sections for their implementation aspects.

3.3. Provisioning, metering, and charge-back management

In the proposed architecture, the resource exchange information is logged at the Unit, Administrator, and EI Private Cloud levels. The Local Server of a Unit logs categorical information of all the connected systems like what type of resources were provided by which system, what was the amount of provided resources by the system, what was the time of resource-sharing, etc. Similarly, the Administrator and EI Private Cloud maintain a log for the resources received from and granted to each Unit. So, the provisioning, metering, and charge-back of the resources can be managed based on the resource contributions done by each Unit and the resource exchange agreement between the Unit and EI Private Cloud.

3.4. Policy Management

The resource exchange policy is a service-level agreement between a Unit and EI Private Cloud. This agreement establishes an equivalence among resources based on their value in the entire EI system. It establishes a basis for metering the use of resources and accordingly makes a charge-back plan to resolve the provisioning and resource exchange issues. For instance, if a policy demonstrates that the deployment server for 30 days is equal to 10000GB storage for 40 days or at $100, then the Unit has to contribute 10000GB storage for 40 days or pay $100 to get the deployment server for 30 days.

3.5. Identity and security management

Uniquely identifying each system in the whole system is mandatory to ensure secure communication. In the proposed architecture, a system belonging to a Unit first qualifies for the security policy, and then it gets a unique identity in the whole EI system. This means a system cannot be a part of the EI system until it clears the identity and security checks at the Unit level. As stated earlier, the Unit also facilitates its connected systems to exchange resources locally through the Local Server, and similar identity and security management are used to ensure local policy compliance.

The proposed architecture is flexible enough to accommodate more EIs and to make a community/shared cloud for more than one EI, regardless of their geographical locations. For instance, if there are N EIs and they agree to build a shared cloud to contribute their excessive resources to the resource pool of the shared cloud, then they can join by accepting the predefined resource-sharing policy. Figure 5 pictorially depicts this idea.

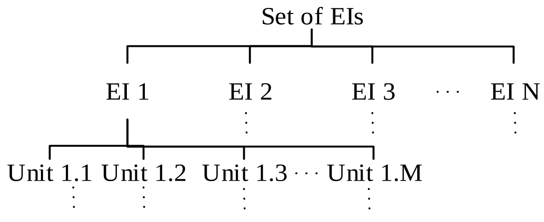

Figure 5. Generalization of the proposed idea for building a shared cloud of multiple EIs

Figure 5 shows N numbers of EIs and each EI can further be divided into its multiple functional Units. In this case, each Unit's Local Server will communicate with its EI's Administrator and that Administrator will communicate with the global Administrator of the shared cloud. The resource-sharing mechanism will remain the same except that (1) the global Administrator will have the authority to add to and grant resources from the global resource pool of the shared cloud. (2) The local sharing of the resources within the EI will be done at the Unit level and the EI level, too, according to the defined local policies at both levels.

3.6. Implementation and deployment feasibility

The cloud is not a new term, but still, most EIs face a bottleneck in managing such a huge infrastructure. Several cloud vendors are providing cloud services, but their fees are so high that low-budget EIs cannot afford them. To overcome this gap, several open-source software tools are available for building private or public cloud infrastructure at a low cost. Among them, OpenNebula, CloudStack, VMware, AppScale, Kubectl, OpenStack, Ubicloud, Terraform, Ansible, Kubernetes, Vantage, Cast AI, etc., are widely used these days. Most of them support the most popular hypervisors like VMware, Oracle VM server, Microsoft Hyper-V, KVM, XCP, Citrix XenServer, etc.

Regarding the robustness and reliability of these open-source tools, there are hundreds of the world's largest brands that are using these technologies to run their business on the cloud. A few of them are CERN, Volkswagen Group, China Mobile, Snapdeal, AT&T, etc. Currently, ample information exists for deciding which open-source tools, technologies and frameworks best suit underlying requirements [38 - 44]. The best mix and match of such open-source tools, technologies, and frameworks will help in economically deploying and implementing the proposed idea.

4. Methodology Dry Run

Suppose "EduIns" is an EI that has one admission office, five faculties, one wet sampling lab, one dry lab for drug repurposing, one maintenance department, one administration office, and a recruitment office. Among all these, the drug repurposing lab is specialized for discovering new drug combinations to treat new disease indications with the help of already discovered drugs [45 - 56]. The further structural division of the drug repurposing lab is shown in Figure 6.

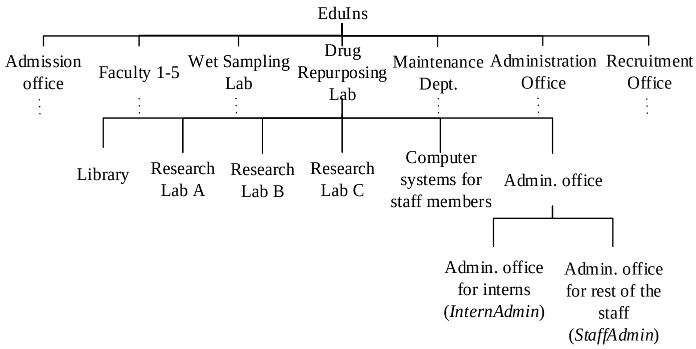

Figure 6. Representation of the "EduIns" EI for its structural constituents.

Figure 6 represents the structural constituents of "EduIns". Among all its constituents, the Drug Repurposing Lab is detailed for its sub-constituents. Similarly, other departments of "EduIns" can also be further divided into sub-constituents. For the sake of simplicity, it is assumed that all the components of the "EduIns" are abstractly defined except the drug repurposing lab. The drug repurposing lab is concretely defined in terms of its structure, shown in Figure 6 and statistical data presented in Table 2. Based on this assumption, the proposed solution is applied to the drug repurposing lab and evaluated for its significance. Similarly, the proposed solution can be applied to the other components of "EduIns" or even to any other EI having statistical data and similar structural components.

The Drug Repurposing Lab has a digital library, three research labs for consultants, around twenty permanent researchers, interns and a local administration staff. The administration staff is divided into two sections, e.g., InternAdmin and StaffAdmin. InternAdmin deals with the affairs of interns, and the StaffAdmin manages the matters of the rest of the staff members. With the help of example data, Table 2 explains the details of existing computation and storage resources in the Drug Repurposing Lab of "EduIns".

Table 2. Details of available resources in the drug repurposing lab.

Sub Units	No. of CSs	ST (GBs)	Avg. ST Need	Rem. ST	Free ST (GBs)	CR (GHz)	Avg. CP Need	Rem. CP	Free CP (GHz)
Library	10	10*150 =1500	20%	80%	1200	10*4 =40	20%	80%	32
Res. Lab 1	50	50*500 =25000	50%	50%	12500	50*4 =200	50%	50%	100
Res. Lab 2	60	60*500 =30000	50%	50%	15000	60*4 =240	50%	50%	120
Res. Lab 3	50	50*500 =25000	50%	50%	12500	50*4 =200	50%	50%	100
Staff comp.	20	20*500 =10000	40%	60%	6000	20*4= 80	40%	60%	48

InternAdmin	8	8*150= 1200	20%	80%	960	8*4= 32	20%	80%	25.6
StaffAdmin	6	6*150= 900	20%	80%	720	6*4= 24	20%	80%	19.2
Total	204	93600	35.7%	64.3%	48880	816	35.7%	64.3%	444.8

In Table 2, ST, CS, Avg., CR, CP, and Rem. Stands for Storage, Computer Systems/Systems, Average, Computational Resource, Computational Power, and Remaining, respectively. Table 2 shows that the Drug Repurposing Lab of "EduIns" has a total of 204 systems, and their approximate collective data storage capacity is 93600 GB, and computational power is 816 GHz. Statistics in the table show that approximately 48880 GB of storage and 444.8 GHz of computation power are freely available. This means that most of the time, such resources are underutilized. In other words, the need of the assignee of these systems is less than the available resources. So, a significant fraction of these extra resources can be made available to EI Private Cloud instead of obtaining other services like remote deployment, web or application servers, analytical tools, etc.

In Figure 6, all the nodes that cannot be further divided can be considered as a Functional Unit or Unit in the whole EduIns. Figure 7 shows the networking model of the Research Lab A of "EduIns".

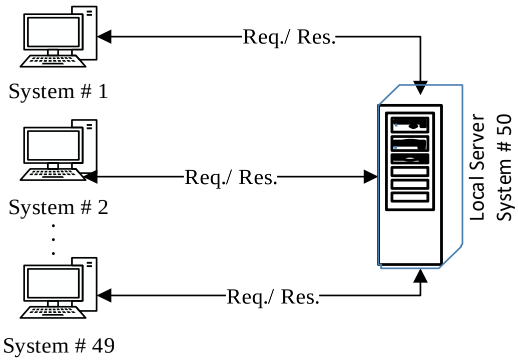

Figure 7. Structure of Research Lab A of the "EduIns".

Research Lab A collectively has 50 systems. One serves as a Local Server, and the rest are clients. System 1 shares 300 GB storage capacity and 3 GHz computation power with the Local Server. Similarly, the rest of the systems also share their free resources with the Local Server. The Local Server gathers all these free resources and maintains information in its resource log file. So, such a resource-sharing model exhibits the Local Server as a small private cloud for Research Lab A.

The log file of the Local Server maintains information on the shared resources based on several parameters like System No, Spare storage, Maximum storage, Maximum computation power, and Spare computation power, etc. Table 3 represents a sample entry of a shared resource in a Local Server's log file.

Table 3. A sample log entry represents that System 1 of Research Lab A provides its free resources to the Local Server.

No.	System. No.	Maximum storage (GB)	Spare storage (GB)	Maximum computation power (GHz)	Spare computation power (GHz)
1	System 1	600	300	6	3

The statistics of log entry in Table 3 show that System 1 has a maximum of 600 GB storage and 6 GHz computational power. Out of the maximum, 300 GB storage and 3 GHz computation power are spare and available for sharing with other systems through the Local Server.

Similarly, the Local Server's log for providing information regarding requested services has parameters like Service requested by, service request time, Service grant time, Service request type (i.e., software, infrastructure, virtual machine, etc.), service period, total cost, adjustment, reason of adjustment, locally granted, total cost to pay, etc. Table 4 shows a sample entry of the server log file.

The statistics given in Table 4 explain that System 1 requested the Deployment Server on 10-Aug-2024 at 11:42 AM for 30 days, and this request was granted on 10-Aug-2024 at 12:43 AM. The Local Server was unable to fulfil the request due to the unavailability of sufficient resources, so it was required to send a service request to EI Private Cloud. The actual cost of this requested service is $1100 for 30 days. As System 1 also provides 300 GB of free storage space for 30 days, so $900 is subtracted from the total service cost. So, the System 1 will pay only $200 for 30 days. The cost of each service, adjustment rate, service exchange, accessibilities or authentication, etc. will all be fixed as per the agreed policy of EI Private Cloud providers. Moreover, if the Local Server fulfils the request of the system, then the service will be granted as per the policy of the Local Server.

Table 4. A sample server log entry that represents a service request and response for System 1 of Research Lab A.

Parameter	Values
No.	1
Service requested by	System 1
Service request time	10-Aug-2024 at 12:42 AM
Service grant time	10-Aug-2024 at 12:43 AM
Service request type	Deployment Server
Service period	30 Days
Locally granted	No
Total cost (In USD)	1100
Adjustment (In USD)	900
Reason of adjustment	Provided free space = 300 GB
Total cost to pay (In USD)	200

The Administrator logs information at the Unit level. For example, all the constituents of Research Lab A collectively constitute a single Unit. So, one of the responsibilities of the Administrator is to maintain a log file containing resource-sharing information categorically about each Unit. Each log entry is based on certain parameters to categorize each data fact for efficient logging, like the Unit number, storage shared, computation power shared, storage sharing duration, computation power-sharing duration, storage grant time, computation power grant time, etc.

Table 5. A sample log entry of the Administrator's log file shows that Research Lab A shares its spare resources with EI Private Cloud.

Parameters	Values
No.	1
Unit number	Research Lab A
Storage shared	13000 GB
Computation power shared	110
Storage sharing duration	31 days
Computation power-sharing duration	31 days
Storage grant time	11-Aug-2024 at 12:43 AM
Computation power grant time	11-Aug-2024 at 12:44 AM

The statistics given in Table 5 detail that Research Lab A shares 13000GB storage resources and 110 MHz computation power to EI Private Cloud for 31 days, and these resources are granted on 11-Aug-2024 at 12:44 AM.

In agreement with EI Private Cloud, the Administrator maintains a resource exchange agreement for sharing resources among Units. This agreement encompasses a resource equivalence policy and the cost of each resource. The Administrator grants services based on this agreement to avoid any resource exchange conflict. Some of the parameters of the agreement are service identifier, type, cost for 30 days, storage granted for 30 days, computation power granted for 30 days, etc. Table 6 explains some sample entries for the service agreement policy.

Table 6. Sample agreement entries to exchange resources among Units.

No.	Service Identifier	Type	Cost for 30 days (USD)	For 30 days	
				Storage (GBs)	Computation power (GHz)

1	Deployment Server	Software	600	1000	3
2	Database Server	Software	300	180	1.5
3	Backup Storage	Storage Unit	200	Not Available	2
4	Graphical Processing Unit	Processing Unit	600	1000	Not Available

The entries in Table 6 show that the Deployment Server will be provided to the requesting Unit either at the cost of $600 or the Unit has to share its 1000GB storage or 3GHz computational power for 30 days with EI Private Cloud. If a Unit does not have any storage or computation power to share, then that Unit will have to pay the cost. Each Unit that joins the EI Private Cloud must agree to proceed with this agreement. The cost and resource exchange agreement will update as new resource types are added or the cost of the resources is changed over time.

5. Conclusion, Limitations, And Future Directions

Reutilization of the existing resources is one of the prime objectives of any software architecture. The inherent nature of the system architectures for cloud-based systems is elastic and service-oriented. EIs opt for cloud services as per their academic research and development needs while keeping budgetary constraints in place. Especially those EIs that face low budgetary constraints, think about the utilization of their existing IT resources. This study proposes a cloud-based software architecture that provides resource sharing and an elastic computing platform while accommodating the existing IT resources of the low-budget EI. Results show that biological research labs (e.g., specialized for drug repurposing) in EIs that require extensive computational and storage powers can easily benefit from the resources that are underutilized in other labs or departments of the EI. Furthermore, the proposed architecture is developed based on structural and requirement similarities among different EIs, and so is a scalable and adaptable framework depending on institutional capacity. Currently, we are working on enhancing it for non-educational sectors like health care, commerce, banks, etc. We aimed to overcome this limitation and are striving to develop separate architectures for each sector based on structural and requirement commonalities among its working bodies. Furthermore, we are working on identifying open-source or non-commercial tools, technologies, frameworks, technology stacks, etc., to implement these architectures cost-effectively.

6. Acronyms and Abbreviations

Table 7 shows the list of acronyms and abbreviations.

Table 7. Acronyms and abbreviations

Acronyms	Abbreviations
IT	Information Technology
EI	Educational Institute
NIST	National Institute of Standards and Technology
PC	Personal Computer
XCP	Xen Cloud Platform
USD	United States Dollar
GB	Gigabyte
GHz	Gigahertz

Funding: Not Applicable.
Data Availability Statement: Not Applicable.
Acknowledgments: Not Applicable.
Conflicts of Interest: Not Applicable.

References

1. P. M. Mell, T. Grance, "The NIST definition of cloud computing," NIST, Sp 800-145, 2011, doi: 10.6028/NIST.SP.800-145
2. M.M. Attaran, S Attaran, G.B. Celik, "Promises and challenges of cloud computing in higher education: A practical guide for implementation," Journal of Higher Education Theory and Practice, vol. 17, no. 6, pp. 20–38, 2017. Available: https://www.articlegateway.com/index.php/JHETP/article/view/1523.
3. M. Thakur, A. Buniello, C. Brooksbank, et al., "EMBL's European Bioinformatics Institute (EMBL-EBI) in 2023," Nucleic Acids Research, vol. 52, no. 1, pp. D10-D17, 2023, doi: 10.1093/nar/gkad1088.
4. V. Lapatas, M. Stefanidakis, R.C. Jimenez, et al. "Data integration in biological research: an overview," Journal of Biological Research, vol. 22, no. 1, 2015, doi:10.1186/s40709-015-0032-5.
5. V. Gligoriević, N. Pržulj, "Methods for biological data integration: perspectives and challenges," Journal of the Royal Society Interface, vol. 12, no. 112, 2015, doi:10.1098/rsif.2015.0571.
6. A. Gital, A. Ismail, M. Chen, et al., "A Framework for the Design of Cloud Based Collaborative Virtual Environment Architecture," in Proc. International Multi Conference of Engineers and Computer Scientists (IMECS), 2014. Available: https://www.iaeng.org/publication/IMECS2014/IMECS2014_pp196-200.pdf
7. M. Mircea, L. Andreescu , "Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis," Communications of the IBIMA, vol. 2011, no. 15, 2011, doi: 10.5171/2011.875547.
8. D. J. Rigden, X. M. Fernández, "The 2024 Nucleic Acids Research database issue and the online molecular biology database collection," Nucleic Acids Research, vol. 52(Database issue), pp. D1-D9, 2024, doi: 10.1093/nar/gkad1173.
9. D. J. Rigden, X. M. Fernández, "The 2023 Nucleic Acids Research database issue and the online molecular biology database collection," Nucleic Acids Research, vol. 51(Database issue), pp. D1-D8, 2023, doi: 10.1093/nar/gkac1186.
10. D. A. Benson, M. Cavanaugh, K. Clark, et al., "GenBank," Nucleic Acids Research, vol. 46(Database issue), pp. D41-D47, 2018, doi:10.1093/nar/gkx1094.
11. J. Mashima, Y. Kodama, T. Fujisawa, et al., "DNA Data Bank of Japan," Nucleic Acids Research, vol. 45, pp. D25-D31, 2017, doi: 10.1093/nar/gkw1001.
12. A. L. Toribio, B. Alako, C. Amid, et al., "European Nucleotide Archive in 2016," Nucleic Acids Research, vol. 45, pp. D32-D36, 2017, doi: 10.1093/nar/gkw1106.
13. F. Belleau, M. A. Nolin, N. Tourigny, et al., "Bio2RDF: towards a mashup to build bioinformatics knowledge system," Journal of Biomedical Informatics, vol. 41, no. 5, pp. 706-16, 2008, doi: 10.1016/j.jbi.2008.03.004.
14. J. Zheng, Z. Xiang, C. J. Stoeckert, et al., "Ontodog: a web-based ontology community view generation tool," Bioinformatics, vol. 30, no. 9, pp. 1340-1342, 2014, doi: 10.1093/bioinformatics/btu008
15. M. Dumontier, C. J. Baker, J. Baran, et al., "The semantic science integrated ontology (sio) for biomedical research and knowledge discovery," Biomed Semantics, vol. 5, no. 1, pp. 14, 2014, doi: 10.1186/2041-1480-5-14.
16. K. M. Livingston, M. Bada, W. A. Baumgartner, et al., "Kabob: ontology-based semantic integration of biomedical databases," BMC bioinformatics, vol. 16, no. 126, 2015, doi: 10.1186/s12859-015-0559-3.
17. L. M. Haas, P. M. Schwarz, P. Kodali, et al., "DiscoveryLink: a system for integrated access to life sciences data sources," IBM Systems Journal, vol. 40, no. 2, pp. 489-511, 2001, doi: 10.1147/sj.402.0489.
18. E. Cadag, B. Louie, P. J. Myler, et al., "Biomediator data integration and inference for functional annotation of anonymous sequences," Pacific Symposium on Biocomputing, vol. 12, pp. 343-354, 2007, doi: 10.1142/9789812772435_0033.
19. Z. B. Miled, N. Li, G. M. Kellett, et al., "Complex life science multidatabase queries," in Proc. IEEE, 2002, pp. 1754-63, doi: 10.1109/JPROC.2002.804683.
20. D. Smedley, S. Haider, B. Ballester, et al. "BioMart - biological queries made easy," BMC Genomics, vol. 10, no. 22, pp. 1-12, 2009, doi: 10.1186/1471-2164-10-22.
21. J. Ostell. "The Entrez Search and Retrieval System. 2002 Oct 9 [Updated 2014 Jan 31]. In: The NCBI Handbook [Internet]. 2nd edition. Bethesda (MD)," National Center for Biotechnology Information, Bethesda, USA, Jan 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK184582/
22. M. Masseroli, A. Canakoglu, S. Ceri, "Integration and querying of genomic and proteomic semantic annotations for biomedical knowledge extraction," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 13, no. 2, pp. 209-219, 2016, doi: 10.1109/TCBB.2015.2453944.
23. R. N. Smith, J. Aleksic, D. Butano, et al., "InterMine: a flexible data warehouse system for the integration and analysis of heterogeneous biological data," Bioinformatics, vol. 28, pp.3163–5, 2012, doi: 10.1093/bioinformatics/bts577.

24. O. Irshad, U. Ghani, “Integration and Querying of Heterogeneous Omics Semantic Annotations for Biomedical and Biomolecular Knowledge Discovery,” Current Bioinformatics, vol. 14, no. 8, 2019, doi: 10.2174/1574893614666190409112025.
25. D. B. Rhee, M. M. Croken, K. R. Shieh, et al., “toxoMine: an integrated omics data warehouse for Toxoplasma gondii systems biology research,” Database (Oxford), 2015; doi: 10.1093/database/bav066.
26. K. Wolstencroft, R. Haines, D. Fellows, et al., “The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud,” Nucleic Acids Research, vol. 41(Web Server issue), pp. W557-W561, 2013, doi:10.1093/nar/gkt328.
27. L. H. Childs, S. Mamlouk, J. Brandt, et al., “Sofia: a data integration framework for annotating high-throughput datasets,” Bioinformatics, vol. 32, no. 17, pp. 2590-7, 2016, doi: 10.1093/bioinformatics/btw302.
28. D. Blankenberg, N. Coraor, G. V. Kuster, et al., “Integrating diverse databases into an unified analysis framework: A Galaxy approach,” Database (Oxford), vol. 11, pp. 1-9, 2011, doi: 10.1093/database/bar011.
29. O. Irshad, U. Ghani, “Formalization and Semantic Integration of Heterogeneous Omics Annotations for Exploratory Searches,” Current Bioinformatics, vol. 16, no. 1, pp. 162 – 178, 2020, doi: 10.2174/1574893615666200127122818.
30. S. Bansal, S. Singh, A. Kumar, “Use of Cloud Computing in Academic Institutions,” IJCST, vol. 3, no. 1, 2012. Available: https://scispace.com/pdf/use-of-cloud-computing-in-academic-institutions-22vp3kavfk.pdf.
31. S. Mokhtar, S. Ali, A. Al-Sharafi, et al., “Cloud Computing in Academic Institutions,” in Proc. 7th International Conference on Ubiquitous Information Management and Communication (ICUIMC), Malaysia , 2013 pp. 1- 7, doi: 10.1145/2448556.244855.
32. S. Al Noor, G. Mustafa, S. Alam, et al., “A Proposed Architecture of Cloud Computing for Education System in Bangladesh and the Impact on Current Education System,” International Journal of Computer Science and Network Security (IJCSNS) 2010; 10(10). Available: http://paper.ijcsns.org/07_book/201010/20101002.pdf.
33. T. Ercan, “Effective use of cloud computing in educational institutions,” Procedia Social and Behavioral Sciences, vol. 2, pp. 938-942, 2010, doi: 10.1016/j.sbspro.2010.03.130.
34. U. Bora, M. Ahmed, “E-Learning using Cloud Computing,” International Journal of Science and Modern Engineering (IJISME), vol. 1, no. 2, 2013. Available: https://www.ijisme.org/wp-content/uploads/papers/v1i2/B0111011213.pdf.
35. B. Nagalakshmi, S. Archana, “Advanced Cloud Computing Technology for Accelerating Innovations in E-Learning,” Middle-East Journal of Scientific Research, vol. 20, no. 3, pp. 343-346, 2014, doi: 10.5829/idosi.mejsr.2014.20.03.11312.
36. M. Masud, X. Huang, “An E-learning System Architecture based on Cloud Computing,” World Academy of Science, Engineering and Technology, vol. 62, pp. 74-78, 2012. Available: https://jmeonline.in/admin/uploadss/1.%20JMEONLINE%20Aditya%20Mittal,%20Gourav%20Saini.pdf.
37. W. Shunye, L. Dayong, Z. Zijuan, “E-Learning system architecture based on Private Cloud for University,” Journal of Chemical and Pharmaceutical Research, vol. 6, no. 5, pp. 492-498, 2014. Available: https://scispace.com/pdf/e-learning-system-architecture-based-on-private-cloud-for-5btj26gy55.pdf.
38. A. Barkat, A. Santos, T. Ho, “OpenStack and CloudStack: open source solutions for building public and private clouds,” in 2014 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2014, pp. 429-436, doi: 10.1109/SYNASC.2014.64.
39. S. Wind, “Open Source Cloud Computing Management Platforms: Introduction, Comparison and Recommendations for Implementation,” in IEEE Conference on Open Systems, 2011, pp. 175-179, doi: 10.1109/ICOS.2011.6079245.
40. L. Voras, B. Mihaljeci, M. Orli, et al. “Evaluating Open-Source Cloud Computing Solutions,” in Proc. 34th International Convention, MIPRO, 2011, pp. 209-214. Available: https://ieeexplore.ieee.org/document/5967051.
41. X. Wen, G. Gu, Q. Li, et al., “Comparison of open-source cloud management platforms: OpenStack and OpenNebula,” In Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, pp. 2457-2461, May 2012, doi: 10.1109/FSKD.2012.6234218.
42. O. Irshad, U. Ghani, “A Comparative Analysis of Biological Data Integration Systems Famous for Data Exploitation and Knowledge Discovery,” Current Bioinformatics, vol. 16, no. 5, pp. 662 – 681, 2021. doi: 10.2174/1574893615999210101125442.
43. O. Irshad, U. Ghani, R. Iqbal, et al., “Performance Optimization of IoT Based Biological Systems Using Deep Learning,” Computer Communications, vol. 155, pp. 24-31, 2020. doi: 10.1016/j.comcom.2020.02.059.

44. M. Z. Khan, U. Ghani, O. Irshad, et al., “Deep Learning and Blockchain Fusion for Detecting Driver's Behavior in Smart Vehicles,” Internet Technology Letters, vol. 3, no. 6, 2019, doi: 10.1002/itl2.119.
45. F. R. Weth, G. B. Hoggarth, A. F. Weth, et al., “Unlocking hidden potential: advancements, approaches, and obstacles in repurposing drugs for cancer therapy,” British Journal of Cancer, vol. 130, pp. 703–715, 2014. doi: 10.1038/s41416-023-02502-9.
46. Y. Xia, M. Sun., H. Huang, et al., “Drug repurposing for cancer therapy,” Signal Transduction and Targeted Therapy, vol. 92, 2024. doi: 10.1038/s41392-024-01808-1.
47. V. S. Kulkarni, V. Alagarsamy, V. R. Solomon, et al., “Drug Repurposing: An Effective Tool in Modern Drug Discovery,” Russian Journal of Bioorganic Chemistry, vol. 49, no. 2, pp. 157-166, 2023, doi: 10.1134/S1068162023020139.
48. T. N. Jarada, J. G. Rokne, R. Alhajj, “A review of computational drug repositioning: strategies, approaches, opportunities, challenges, and directions,” Journal of Cheminformatics, vol. 12, no. 1, pp. 46, 2020. doi: 10.1186/s13321-020-00450-7.
49. N. Krishnamurthy, A. A. Grimshaw, S. A. Axson, et al., “Drug repurposing: a systematic review on root causes, barriers and facilitators,” BMC Health Services Research, vol. 22, no. 1, pp. 970, 2022, doi: 10.1186/s12913-022-08272-z.
50. A. H. Jonker, D. O. Daniel, M. C. Bellaubi. et al., “Drug repurposing for rare: progress and opportunities for the rare disease community,” Frontiers in Medicine, vol. 11, 2024, doi: 10.3389/fmed.2024.1352803.
51. M.K. Moka, M. George, D. Rathakrishnan, et al., “Trends in drug repurposing: Advancing cardiovascular disease management in geriatric populations," Current Research in Translational Medicine, vol. 73 (2), 2025, doi: 10.1016/j.retram.2025.103496.
52. D. Hong, A.S. Kesselheim, "Strategies to Advance Drug Repurposing for Rare Diseases," JAMA Network Open, vol. 8, 2025, doi: 10.1001/jamanetworkopen.2025.8339.
53. C.V. Karnebeek, A.R. Müller, L. Benkemoun, et al., "SIMPATHIC: Accelerating drug repurposing for rare diseases by exploiting SIMilarities in clinical and molecular PATHology," Molecular Genetics and Metabolism, vol. 144 (4), 2025, doi: 10.1016/j.ymgme.2025.109073.
54. S. Nijim, A. Korsunska, J. Zinski, et al., "Rare Disease Drug Repurposing," JAMA Network Open, vol. 8 (5), 2025, doi: 10.1001/jamanetworkopen.2025.8330.
55. P. Czechowicz, A. Więch-Walów, J. Sławski, et al., "Old drugs, new challenges: reassigning drugs for cancer therapies," Cellular & Molecular Biology Letters, vol. 27, 2025, doi: 10.1186/s11658-025-00710-0.
56. S. Chaudhary, A. Rana, S. Bhatnagar, "Drug Repurposing: A Conduit to Unravelling Metabolic Reprogramming for Cancer Treatment," Mini Reviews in Medicinal Chemistry, vol. 25 (8), pp. 601-627, 2025, doi: 10.2174/0113895575339660250106093738.