## Legal Informatics

*Research article*

# *eLegalls-as-a-Service*: Towards Developing Cloud-based Legal Tech System to Aid Lawyering in the Digital Age

**Sugam Sharma**

Iowa State University (ISU) Startup Factory/eLegalls.com, 1805 Collaboration PI, Ames, Iowa, USA 50010,

info@elegalls.com, sugam.k.sharma@gmail.com, ORCID: 0000-0002-0762-3900

**Ritu Shandilya**

Mount Mercy University, Cedar Rapids, Iowa, USA 52402,

rshandilya@mtmercy.edu, ORCID: 0009-0000-8205-9302

**Divya Dwivedi**

Supreme Court of India, 40 Hanuman Lane, Connaught Place, New Delhi, India 110001,

divyadwivedi04@gmail.com, ORCID: 0000-0002-6086-7090

**Millie Pant**

Department of Applied Mathematics and Scientific Computing, Indian Institute of Technology Roorkee, India 247667,

pant.milli@as.iitr.ac.in, ORCID: 0000-0002-7668-7887

## Abstract

Legal Tech Jurisprudence is not as developed as it was hoped for, compared with other fields of study. As a result, legal tech has not advanced globally as would have been preferred and remains primal in nature. The situation is more disappointing when it comes to developing or underdeveloped countries. While tech is critical in countries like India, which is going through digital transition currently, the justice system is still functioning in centenarian ways. For some reason, the system has not yet begun to fully harness the potential of modern IT technologies, which may consist of AI, ML, DL, NLP, etc. but not limited to these technologies. A lawyer, who is an integral part of the justice system, still continues to handle most tasks manually or with the help of an assistant, which often becomes challenging and cumbersome while dealing with complex legal issues that involve humongous contracts, for example. It is problems like these that can be very well handled with the help of technologies, being dubbed as legal informatics (LI), to help elevate the quality and quantity of lawyering and the core of jurisprudence. To try and resolve this problem and find an amicable solution to an extent, we have designed and developed an LI-enabled and proposedly cloud-based innovative computational system, called *eLegalls* and to be delivered as *eLegalls-as-as-Service*, and this paper illuminates and elaborates its potential in providing the hassle-free lawyering in digital age.

## Keywords

## Introduction

Legal Tech Jurisprudence has not started to evolve as one would have hoped for in comparison to other domains of law. Emerging technologies have time and again proved their mettle by serving humans and benefiting humanity as well. Field that has the most impact on our daily lives is Law and that fraternity has ignored the technologies largely till date. After pandemic hit, there was a

huge hue and cry for usage of virtual courts in many countries and this was not only from practicing lawyers, but also from judiciary too. One cannot agree more than ever that legal court system needs to change its functioning on day-to-day basis and must involve as much technology as possible. Lawyers will have to come to terms with technology as a helping mechanism and not an enemy of sorts. In the past, scientific communities, along with legal scholars and computer scientists, have initiated communications to converge together their isolated expertise and efforts to energize a judicial system with the potential and power of modern technologies like Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and Natural Language Processing (NLP), etc.; and, concept of applying modern technologies in legal domain is termed as Legal Informatics (LI).

As evolution of LI and LI-enabled models has been relatively slower and, therefore, it requires swift advancements to upgrade and to eventually become part of Legal Tech Jurisprudence. In developing countries with large population, like India, technological urgency to enhance the legal system with LI capabilities is more manifested. In Indian Legal System, court proceedings are very slow and number of cases are very high, and as a result, the decision-making process to react final verdict often takes too long, sometimes decades. It has been often reported that the Indian Justice System lacks requisite number of judges to adjudicate current, pending and piled up court cases. Although, the argument may be true in nature, but it does not help towards the betterment of system. Hence, possibly by the motivation being manifested by the success of technological advancements in other fields of study, the legal community has started exploring the possibilities of usage of technology in the system of law (LI) and, therefore, some thin traces of LI are witnessed in literature today. Currently, some of the visible LI developments globally include the conversion of paper form of legal documents, such as court proceedings and cases into digital form, research and development (R&D) of AI exploration in jurisprudence aiming to produce efficient software tools to search and extract digital documents from legal corpus; a few legal software tools are aiding, particularly, legal scholars and experts in reporting searched cases in a graph with their citations as nodes and vertices, respectively. Although, the development of LI is slow, any major change is yet to be seen where law and justice stand firm with its expectations and promises to be faster, fair and financially feasible, even for the feeble and frail fraternities.

The legal court system primarily consists of three main players — judge, lawyer, and parties involved in litigation. The lawyer is an integral,

indispensable and intrinsic part of a judicial system and is the primary and only connection between judge and parties; lawyers also share the ethical responsibilities of ensuring smooth proceedings and fair justice delivery. Although, each aspect of jurisprudence is in need of urgent and swift technological intervention to ameliorate them; however, the efforts in this paper are focused mainly on improving the lawyering aspect and to facilitate lawyers in their case-related activities. In this digital era, a majority of lawyers or attorneys still deal with their cases on paper, overflowing their dockets. Management of these long documents, and also dockets, often is a challenge for the lawyers and always involves additional unproductive time consumption, which also somewhere directly or indirectly contributes to the delay in justice delivery. There is a viable possibility and a good scope of improvement on this front and in this research work, we provide a technology-enabled computational solution, called *eLegalls* (prototype), for lawyers so they can manage and handle their cases efficiently. The *eLegalls* is architected to be lawyer-centric, equipped with several pertinent features and functions that are highly beneficial to lawyers in their case management activities and effectively discharging their duties in an efficient manner. To make *eLegalls* more robust, secure, affordable and easily accessible, we further leverage another popular service delivery technology, cloud computing, and propose to deliver *eLegalls* as eLegalls-*as-a-service (eLaaS)*.

The remaining paper is structured as follows. Literature on contemporary LI research is compiled in Section 2. Section 3 briefly explores basic cloud computing technology and also outlines its expansion into legal context. Section 4 illustrates architecture of *eLegalls*, with focus on its lawyering core. In Section 5, technical development and implementation of *eLegalls* prototype are illustrated with pertinent test cases. Section 6 discusses current legal tech landscape and outlines some interesting LI-enabled emerging legal techs along with their classification, including *eLegalls*.

## 1. Literature Review

J.B. Ruhl et al. consider legal systems are complex systems and the use of complexity science can help produce better and improved legal system [Ruhl J.B. et al., 2017: 1377−1378]. The study of conventional legal system is majorly based on empirical observations with unnoticeable scientific experimentation and has not explored various perspectives of complexity

science yet. The complexity of legal system is bound to grow in near term, which consequently will push legal industries to investigate new technological solutions and support and to expand technology footprints in legal domain. The complex legal systems always produce complex datasets with distinctive compositions, for example- case decisions, executive orders, legal regulations, legal contracts, legal records, legal opinions, networks and citation networks of cases (Figure 1), etc. Big data science, in combination of modern IT and its allied AI, ML, DL, and NLP technologies, is expected to reshape the traditional legal studies and to improve skills and pragmatic understanding, which ultimately will ameliorate and upgrade the quotidian legal operations. Complexity science when introduced into legal research is anticipated to open avenues to address intrinsic legal questions and concerns, that may help elevating and enhancing a judicial system and refining policy-making to match the societal expectations from legal system. Goswami's thesis on LI primarily explores the information retrieval of legal data, which is dubbed as legal information retrieval (Legal IR). Legal IR retrieves legal documents, for example, legal judgments, case laws, acts, articles, etc. Legal documents are generally exceptionally lengthy and also are written in complex legal language with long sentences; lawyers and other legal experts often require to refer these court documents in their work-related activities. However, legal language complexity and convoluted lengthy legal sentences make their comprehension tedious and tardy. To address this issue, Goswami relies on supervised algorithm and created an auto catchphrase retrieval mechanism for legal content to draw catchphrases out of complex court judgement to impart the gist to quickly understand a judgment. LegalIR runs on database of court judgments,
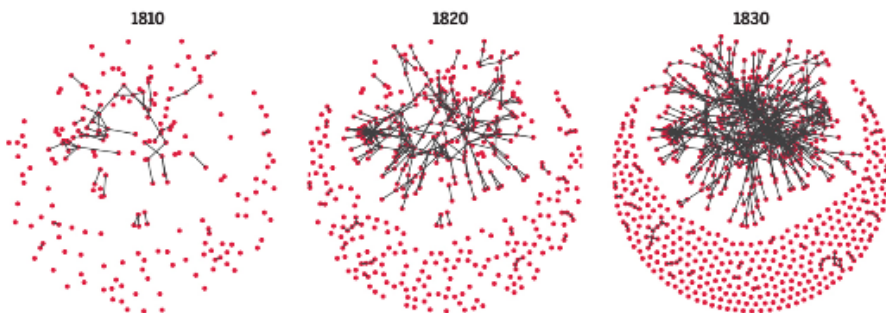


*Fig. 1.* United States Supreme Court citation network (1805−1835) (Ruhl et al., 2017). Nodes are the Cx(s) and the edges are citations of the Cx(s). The Cx(s) citation span is (1791−2015)

*Fig. 2.* Indian Kanoon search engine

*Source:* https://indiankanoon.org/advanced.html.

acts, legal articles, etc. and outsmarts Indian Kanoon (Figure 2) and Manupatra (Figure 3), the online search engines for Indian jurisprudence, in efficiency and accuracy [Goswami J.J., 2018]. Lettieri et al. investigate the possibilities and potential benefits of applying computer visualization in legal domain and called it Visual-LI [Lettieri N., Malandrino D., 2018: 241−246]. Know Lex [Lettieri N., Altamura A., Malandrino D., 2017a: 332−345] (Figure 4), which is equipped with several efficient features, provides the thorough visual analyses of legal documents through its online web interface. The objective of development of Know Lex is to provide a testbed to legal community to study and understand relationships of heterogeneous legal documents. EU Case Net [Lettieri N., Altamura A., Faggiano A., Malandrino D., 2016: 56] (Figure 5) can be considered as a web-based testbed to extensively support to conduct study to develop new tools and techniques for analytics on legal corpora. The objective is to create an online laboratory for legal community to study legal science and legal system in Europe. The study examples include, better understanding of case network of the European Court of Justice (ECJ) of law, conducting visual analytics on ECJ judgements to deduce relevant observations. Crime Miner [Lettieri N., Malandrino D., Vicidomini L., 2017b: 32−54] (Figure 6) equips investigators to find innovate techniques to study and explore crime and its societal implications. Crime Miner offers mining, visualization and analytics operations and SNA techniques to investigate and understand
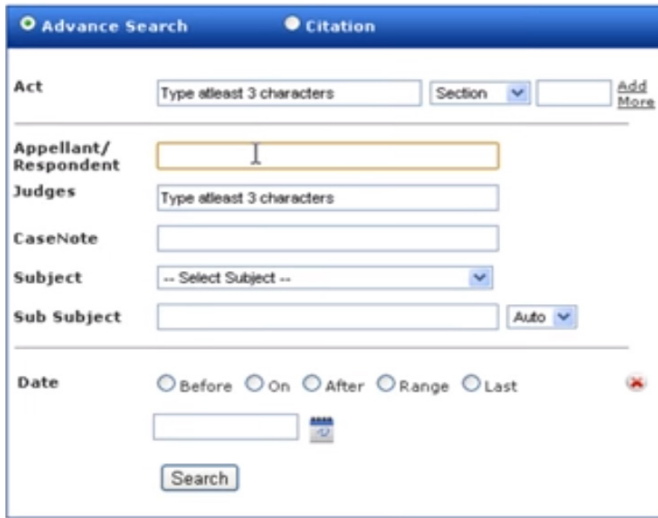
*Fig. 3.* Manupatra online legal research

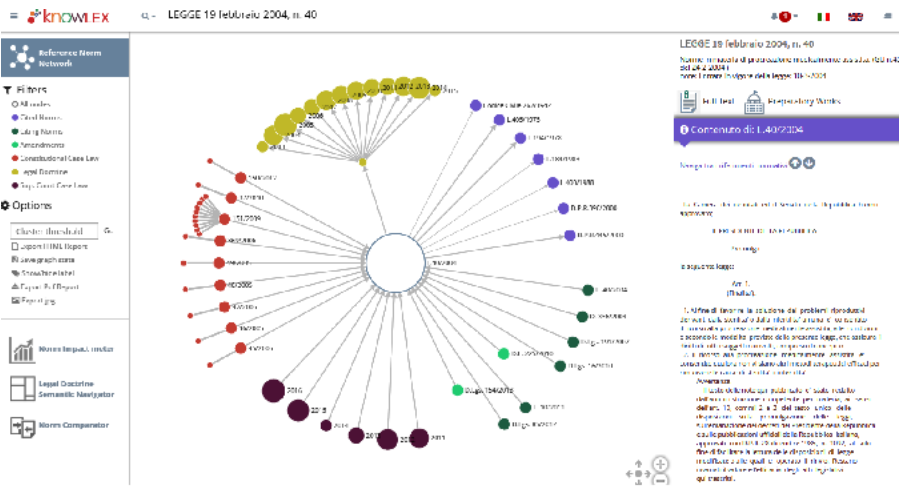Source: http://www.manupatrafast.com.



*Fig. 4.* KnowLex: the Norm Graph Navigator

behavioral and functional patterns of notorious individuals or entities from crime database, which includes phone tapings and personal records with former represents edge and latter is a node or vertex of a graph in SNA data analysis to understand and manifest the connection of individuals or entities complicit in phone conversations. Typically, legal documents are generated
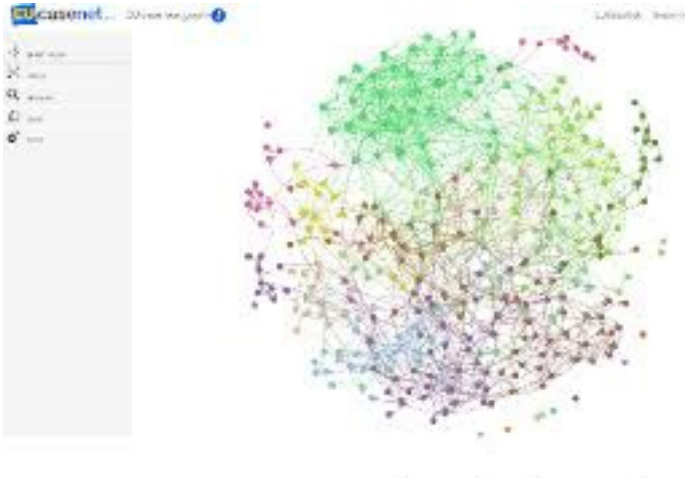
*Fig. 5.* EUCaseNet: Citations network graph of the first 350 judgments in terms of in-degree and their citation network
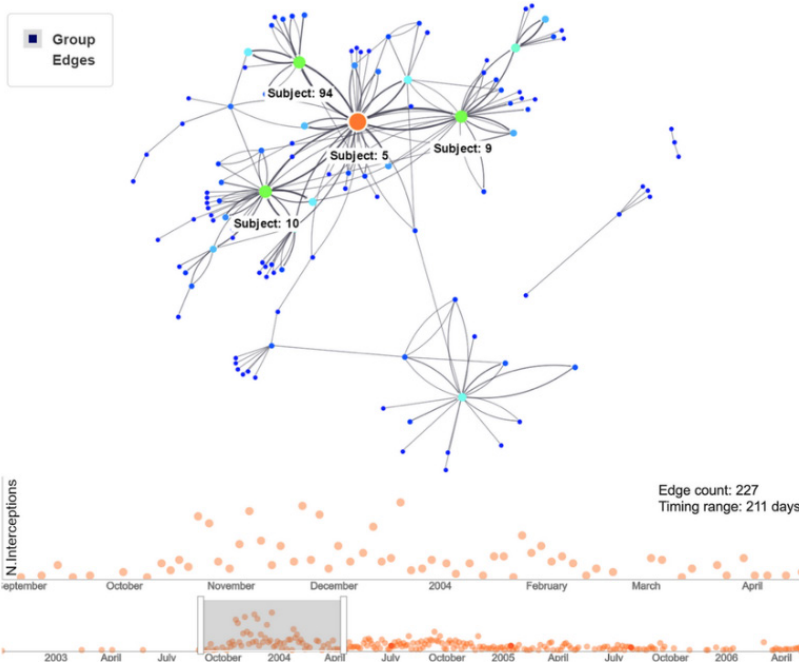


*Fig. 6.* CrimeMiner: The wiretaps graph

in a regulated standard format with smaller font size and thin line spacing and are excessively lengthy. Also, the language of written legal documents is highly technical, flooded with legal jargons, and consequently, is beyond

easy comprehension of an ordinary educated individual with no prior legal understanding. The words and expressions in legal documents are interpretable to various meanings, which potentially change legal discourse and consequently the outcomes. As a result of above, a reader does not fully understand a legal document and also legal communication easily and often requires support from legal experts. Therefore, the notion of design patterns from the IT world is injected into legal document design as an important instrument. Design patterns providing solutions to legal design issues, help to develop legal documents in various simple formats, highly suitable for ordinary readership with no legal domain knowledge. In recent times, several good legal design patterns are introduced, but are underused mostly, probably because of the missing central legal design patterns repository; consequently, searching for a pertinent design pattern becomes a tedious process. Therefore, in the view of rehashing and tailoring complex legal documents to be coherent and comprehensible to a specific audience, A. Rossi et al. have created a design language to restructure the legal design patterns into a desired format. The goal is to gather and accumulate the dispersed, disorganized, and diverse legal resources and solutions and then to attach relevant examples and contextual descriptions to them to improve their utilization and to smoothen the legal communications throughout in legal domain [Rossi A., Ducato R. et al., 2019: 517−526]. Waltl et al. are the first few investigators, who study the possible use of AI in LI, more specifically, the explainable AI, which potentially can enable tagging of proper explanations along with the court decisions in a judicial process. The authors suggest that explainablility in AI can be an additional intrinsic category of ML algorithms, see: [Ruhl J. B. et al., 2017: 1377−1378].

## 2. Cloud Computing in Legal Space

In recent years, cloud computing has emerged as a powerful and revolutionary service-oriented delivery paradigm, primarily because of its on-demand and abstract architecture for complex, large-scale, and data-intensive applications [Alam T., 2021: 108−115]; [Sharma S., 2015]; [Varghese B., Buyya R., 2018: 849−861]. The on-demand nature of cloud paradigm makes it highly cost-effective to users; and also, as cloud environment hides the software-supported complex hardware infrastructure and its cumbersome and overwhelming technical details from the end users, this together encourages them to avoid establishing an expansive local server-level hardware infrastructure and rather offload their complex

applications into cloud. The cloud services are still broadly classified into three traditionally categories:

*Infrastructure-as-a-Service (IaaS)*. This is an important paradigm of cloud environment, which is being used heavily in practice globally. The IaaS model provides the complex infrastructures as simple abstract services to the end users. The enterprise IT infrastructures are constituted by various several comprehensive software, hardware and network resources organized in a more modular fashion; and this modularity facilitates easy customization of the environment to accommodate and support the varying needs of the end users. Amazon Web Services (AWS) [Wittig M., Wittig A., 2018] is the most popular and widely used commercial cloud IaaS model today.

*Platform-as-a-Service (PaaS)*. As name suggests, PaaS model facilitates and provides a secure, mature, and vigorous but flexible platform to the users, where they can develop software systems. The platform is furnished with all the necessary software artifacts; platform itself can either be housed in a localized hardware environment, belonging to a PaaS provider or be served through IaaS. A PaaS cloud can be accessed and used either by PaaS specific APIs or GUI. Google App Engine [Python J., Go P., 2015: 35] is one of the most popular PaaS cloud platforms.

*Software-as-a-Service (SaaS)*. SaaS is a highly popular and most used resource-rich and repository-intensive cloud paradigm, which provides the software to end users in the form of service. The SaaS repository is composed of a sea of diverse sets of simple and complex software services, which are readily available to use. Google [Geewax J. J. J., 2018] as SaaS provider is providing several software as SaaS service, for example Gmail, Google Doc, etc. A SaaS provider owns up full responsibilities to deliver operational services and also its underlining intertwined hardware and software infrastructure. Most SaaS vendors also support their users with efficient GUIs to offer easy and smooth interaction.

Because of on-demand nature, easy accessibility and integration, flexible and affordable (pay as you) pricing model, the cloud computing acceptability has grown across almost all the domains such as healthcare, medicine, business, data science (*Data-as-a-Service*, *Data Integrity-as-a-Service*, *Data Mining-as-a-Service*, *Database-as-a-Service*) [Sharma S.,

2015], etc. In legal domain, most of the discussion is still centered around the legal issues triggered due to inception and evolution of cloud computing itself as disruptive technology such as data (in cloud) privacy and security violation, data breach and loss, hacking, copyright infringement liability, etc. [Morningstar Law Group, 2015]. The cloud data and service usage across cross-borders has further added the legal complexity such as who owns the data and service, who has legal jurisdiction over the cloud data and services, etc., [Reingold B., Mrazik R., D'Jaen M., 2010: 1-6]. Nations are developing new rules, regulations, laws, litigations, and jurisdictions to address the above-mentioned legal issues. The United States has now enacted and established the CLOUD Act, which enables it (with other foreign countries and vice-versa) to lawfully access the data, stored off-shores, required on court orders [Mulligan S. P., 2018: 3−28].

Although, the LI space is still evolving [Sharma S., Gamoura S., Prasad M.D., Aneja A., 2021: 218−235], however, with its evolution, the cloud-usage in legal world is gradually expanding; more and more legal entities are growing and adopting cloud environment for their service and legal data delivery [Corrales M., Fenwick M., Haapio H., 2019]; [Murley D., 2009: 101]; [Dykstra J., Riehl D., 2012: 1]; [Black N., 2013: 593]. And this motivates us also to develop *eLegall*s in a way, which could be easily and smoothly deployed and served from cloud environment as an effective SaaS model and we propose to deliver it as *eLegalls-as-a-Service.*

## 3. Design and Architecture

In this section the design and architecture aspects of the system are illustrated with special focus on lawyering support of *eLegalls*. Various elements, integrated with the system are also depicted. Also, formal definition is given to core constituents of *eLegalls* from the theory building point of view and theirs abbreviation are summarized in Table 1 [Sharma S., AL R. S., 2021:16−31].

*Table 1.*        Abbreviation of *eLegalls* terms

| S. No. | Term | Abbreviation |
|--------|------|--------------|
|  | Victim | $V_L$ |
|  | Complainant or plaintiff | $CP_L$ |
|  | Case, complaint or report | $C_x$ |

| S. No. | Term | Abbreviation |
|---|---|---|
| | Litigant | $L_L$ |
| | Lawsuit | $L_x$ |
| | Accused or defendant | $AD_L$ |
| | Police or law enforcement agency | $P_A L$ |
| | Lawyer, advocate or attorney, litigator | $L_A L$ |
| | Judge or magistrate | $J_L$ |
| | Judiciary, jurisprudence, judicial system, or court of law | $J_C L$ |

1. Victim ($V_L$) is defined as an individual, who experiences suffering as a result of the action of someone ($AD_L (s)$).

2. Complainant or plaintiff ($CP_L$) is defined as any individual who reports the case against the $AD_L (s)$ in a $PA_L$ or $JC_L$.

3. Case/Complaint ($C_x$) is defined as the written document deposited to the $PA_L(s)$, with the detailed information about the event that agonizes $V_L$ or $CP_L$.

4. Litigant ($L_L$) is associated as a party in a $C_x$ as $V_L$, $AD_L$, or $CP_L$.

5. Lawsuit ($L_x$) is dubbed as $C_x$ submitted to the $JC_L$ by a $CP_L$ for adjudication or arbitration.

6. Accused or defendant ($AD_L$) is someone -individual, group, or entity- who stride to defend themselves in a $JC_L$ on the $C_x(s)$ against them.

7. Police or law enforcement agency ($PA_L$) is the government-run entity, which is responsible to administer, implement and enforce the good laws and order.

8. Lawyer, advocate or attorney, litigator ($LA_L$) is someone, who has undergone rigorous educational training to study and understand the laws and the legal system and subsequently is authorized to practicing the law. The $LA_L(s)$ advocate and argue in a $JC_L$ with the aim to derive the decision and final outcome in favor of their clients.

9. Judge or magistrate ($J_L$) is someone, who is equipped with government-enabled authorities, responsibilities, and power to adjudicate the $C_x(s)$, presented in the $JC_L$ before him/her and delivers the fair judgements eventually in $JC_L$.

10. Judiciary, jurisprudence, judicial system, or court of law ($JC_L$) is a system, which consists of the people with judicial understanding and government-enabled authority and power to resolve the $C_x(s)$ or $L_x(s)$ and other disputes through the fair adjudication and arbitration.

Figure 7 depicts the architecture of *eLegalls-as-a-Service*. The core components of system are *eLegalls* and $LA_L$ *(s)* and Law Firm(s). It can easily be seen that *eLegalls* is hosted in cloud along with its dedicated *Data Store*, which consists of (relational) *Database* and *Files* storage in the cloud as well. The *Database* is relational in nature and contains relevant textual data. The *Files* storage houses the large legal documents and the information to access these documents are stored in tables of relational *Database* object. The architecture also shows abstraction of prominent functional services, offered under *eLegalls-as-a-Service* — 1) *Register new lawyer,* 2) *Update case status*, 3) *Case status*, and 4) *Create a new case*.
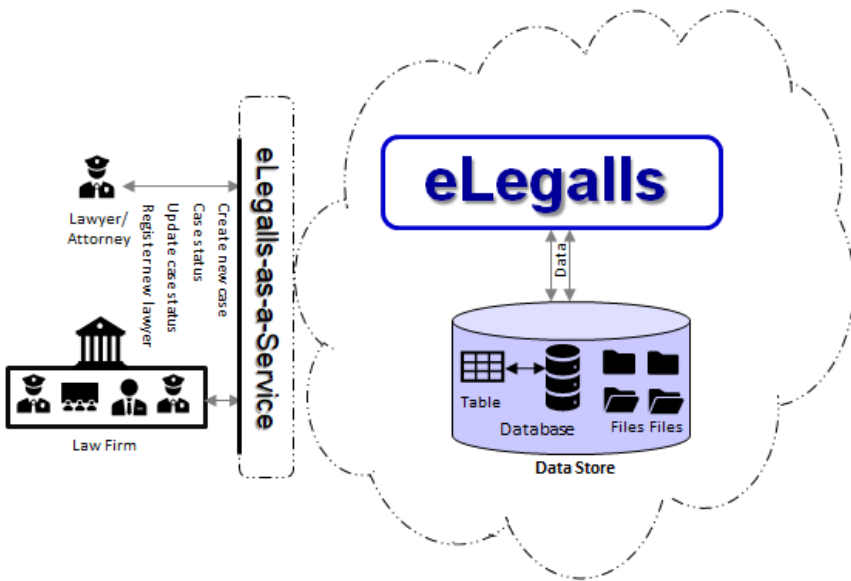


*Fig. 7.* Architecture of eLegalls-as-Service. In this figure, it can be easily observed, the eLegalls system resides in cloud and is served as eLegalls-as-Service to external world, particularly, lawyers/attorneys or law firms. The Data Store consists of Database and Files storage components, which store the related legal data in relational format (tables) and files such as legal documents

## 4. Development and Implementation of *eLegalls* System

Figure 8 is the design mockup of *eLegalls*, which primarily consists of following modules - *Report new case*, *Case status*, and *Lawyer*. Although, similar to the complex nature of legal system, the design of *eLegalls* in its entirety is quite complex, but for simplicity and easy understanding, a basic section is shown here. As mentioned earlier, in this paper, the focus is to illuminate the lawyering core of *eLegalls,* therefore, in this design image, the *Lawyer* module is shown selected and active, where a $LA_L$ (with dummy name "*R J Malani*") is logged in. After successful logging, $LA_L$ is able to perform several tasks such as "*Add new comments"* to update case status of an existing $C_x$, "*Upload new file"* to upload new legal documents, etc.



*Fig. 8.* Basic design mockup of eLegalls

In the following section 5.1, we further explore the implementation aspect of lawyering-centered basic components of *Lawyer* module; the demonstration of each item is supported by a pertinent use case scenario. And thereafter, in section 5.2, we briefly explain the technologies used for

the implementation; also highlighted in this section is the data model, which suffices current data engineering needs and also will be potentially able to accommodate the scalability and high availability system requirements.

### 4.1. Lawyer module implementation

*eLegalls's Lawyer* module is highly beneficial and immensely useful to $LA_L(s)$ in their $C_x$ management and to help them better prepare with the organized and updated documents to effectively aid their lawyering activities and advocacy in $JC_L$(s). The *Lawyer* module consists of a few submodules and each one of them is independently development and implemented.

### 4.1.1 New $LA_L$ to eLegalls

To operate and avail the *eLegalls* system, the new $LA_L$ has to register with it for the first time through a simple but secure user interface (see Figure 9). During the registration process, the system requires the $LA_L$ to enter pertinent information such as *Bar ID*, *Expertise*, etc. along with the important documents, which are to be uploaded. Upon the successful registration, the system sends a confirmation auto email to the $LA_L$, which also carries the confidential login credentials. The accuracy of the $LA_L$'s *Expertise* parameter is crucial to the system as this information is later used to match the $C_x$ types to suitably assign it to the most appropriate $LA_L(s)$. Presently, the *eLegalls's* prototype functions with some basic authentication mechanism, however, as the prototype advances and expands into a mature product, a more sophisticated and robust authentication process such as AD, LDAP, etc. will be integrated into the system [Vazquez A., 2019: 123−155].

### 4.1.2 Existing $LA_L$ login

Figure 10 is the user interface developed in this module; this enables a registered $LA_L$ for the successful login. The interface requires the $LA_L$ *to* input the correct login credentials, which were sent to him/her through the auto email during the first-time registration process. In case, if a $LA_L$ forgets the credentials or has not yet registered with the system, this interface also provides links to navigate a $LA_L$ to other appropriate interfaces.

### 4.1.3. Case status update by $LA_L$

The module is quite important and useful for $LA_L(s)$ when it comes to management and updating of a $C_x$ in *eLegalls* system. Figure 11 shows that

F*ig. 9.* Lawyer (New LAL registration). This sample example demonstrates, how a new LAL can register themselves with eLegalls system. After the successful submission, the system sends an auto email to the LAL with the login credentials



*Fig. 10.* Lawyer (Existing LAL login) - This interface assists the LAL to login using the valid credentials.

a new $C_x$ is created and system has assigned an auto generated unique $C_xN$ (L87020445290010) to this $C_x$. This $C_xN$ is used to surf and search this $C_x$ throughout all the use case examples.

On successful login (in Figure 10), a $LA_L$ is navigated to graphical interface, which facilities $LA_L$ to retrieve the entire most up-to-date case history and to further update $C_x$ status. Figure 12 provides a use case example of a $C_x$ update scenario. The $LA_L$ enters $C_xN$ (L87020445290010) into the designated field and a single click on "Go" button renders entire $C_x$ status. The $LA_L$ decides to update a $C_x$ status and adds more content to it. The newly updated content becomes immediately available in $C_x$ status display with the valid $C_xN$ and this example scenario is shown in Figure 13.

### 4.2. Frontend and backend technologies

The *eLegalls* prototype is developed in modular design using modern computer technologies. The frontend and backend of the system are



*Fig. 11.* Create a new case in eLegalls system- This interface enables the process to generate a new Cx. On the successful submission of a Cx, the eLegalls generates a confidential and unique identification number (CxN) for a Cx. This CxN is tagged with a Cx and is used as a reference for that Cx

**145**

*Fig. 12.* Lawyer (Case status update (LAL)). A click on Go button with a valid Cx number entered, renders the textbox to update the Cx status. The authorized LAL can enter the long comments and clicks on the Update button.



*Fig. 13.* Lawyer (Case status update (LAL)). A click on the Update button prepends the new comments to the Cx status list. The newly updated comments are readily available to the LAL and CPL or a VL (with valid Cx number) tout de suite to view and read.

implemented using the following programming languages — J2EE 1.8, Servlets, Jsp, Jspf (Java server pages fragments to improve code security), Html 5, JQuery, and CSS. The project is developed, built, integrated and tested in Eclipse Juno integrated development environment (IDE) [Araujo R.F., 2013]; [Hickson I., Berjon R., Faulkner S. et al., 2016]. In *eLegalls* software prototype development, modular design is further enforced by enabling and implementing efficient software design patterns such as DAO [Ngaogate W., 2020: 27−31]. Additionally, to have clean and compact code in development process, annotation-based codebase programming is preferred wherever possible [Guérin S., Polet G., Silva C. et al., 2021]. The data engineering, management and data transition in *eLegalls* system are facilitated through SQLite relational data model [Oh G., Kim S. et al., 2015: 1454-1465].

## 5. Discussion on Legal Tech Landscape

One of the primary objectives of emerging LI research and innovation is to alleviate deeply rooted and globally pervasive manual activities in jurisprudence and to exploit the unbounded benefits of the IT and its allied modern technologies to achieve it. More specifically, evolving LI has great potential to develop the manual legal and litigation procedures, practices, and proceedings into an efficient and automated civil justice system with some degree of intelligence. AI-supported LI systems are expected to predict $C_x(s)$ outcomes with high degree of accuracy and therefore, can aid speeding up decision delivery of $C_x(s)$, when augmented with $J_L(s)$ and within $JC_L(s)$. Furthermore, these predicted outcomes at early stage can be immensely advantageous to $CP_L(s)$, $AD_L(s)$, and $LA_L(s)$; specifically, $LA_L(s)$ can repeatedly adjust the input parameters and produce more refined predictions and eventually achieve the most beneficial outcome to their $C_x(s)$. As a result, the most optional final input parameters can be used as supportive factors by $LA_L(s)$ to defend and substantiate their arguments in $JC_L$ to improve, enhance and strengthen the winnability of their $C_x(s)$. In their recently published overarching work on legal tech and its impact on civil procedure, Engstrom and Gelbach have extensively elaborated on current status and future evolution of legal tech [Engstrom D.F., Gelbach J.B., 2020], which is the basis and motivation of this discussion that is primarily centered around the emerging legal techs. In recent years, LI research and LI-enabled legal techs have slowly begun to develop their minuscule footprints; however, substantial LI research presently is still

practically futuristic, formulaic, and foreseeable with remote imagination to realization of robot-judges [Markou C., 2017]; [Volokh E., 2019] and robot-lawyers [Fitch A., 2020]; [Merchant G., Covey J., 2018]; [Markovic M., 2019: 325]. As, LI research inevitably expands, LI-enabled legal tech space inexorably expands, and as a result, the scope of disruptive stress by the legal tech evolution on existing jurisprudence also expands; which further raises some critical questions about adaptability of emerging LI-enabled new legal tech world, filled with unbounded technical challenges, by $JC_L$'s active players, particularly, $J_L(s)$ and $LA_L(s)$; additional obvious concerns are related to the momentum of adoptability, how swiftly, these legal actors are able to accommodate the convoluted complexities and tedious intricacy of new legal tech, when new modern legal tech world begins to rejig, reorganize, and revolutionize the existing civil procedure and civil justice system. The growth of LI-enabled legal tech arena is expected to affect all major legal players of $JC_L$ and strikes on two key challenges- overly lengthy legal procedure and proceeding's time which often runs through years to decades, and extremely unaffordable financial stress, incurred and imposed due to $JC_L$ proceedings.

For better understanding the entire legal tech space, it is advisable and important to segregate them into an appropriate category based on the nature of operations. Therefore, in this section, the emerging legal tech world is further fragmented into a coarser classification.

Legal matchers assess the given expertise of $LA_L(s)$ and aid matching the perspective $CP_L(s)$ to $LA_L(s)$. Additional support to $LA_L(s)$ includes basic operations such as maintaining legal records of $C_x(s)$. Examples — Ravel[1], Lexicata[2], Atticus[3], Avvo[4], etc.

Legal searches aid $LA_L(s)$ in smooth and easy searching of legal documents of their interest, e.g. case decisions, legal proceedings, statutes. Examples are: Judicata[5], Case Text[6], etc.

Legal decision predictors auto predict probable decision of $C_x(s)$ or $L_x(s)$ with a good degree of accuracy. The end users, particularly $LA_L$(s), can

---

[1] home.ravellaw.com (accessed: 20.05.2022)

[2] clio.com/?cta=lexicata (accessed: 16.12.2023)

[3] atticus.com (accessed: 20.05.2022)

[4] avvo.com (accessed: 29.12.2023)

[5] judicata.com (accessed: 16.12.2023)

[6] casetext.com (accessed: 12.04.2024)

generate various outcomes of $C_x/L_x$, based on varying inputs; and, can exploit such input/output combinations to properly formulate, drive and strengthen their future legal arguments for the same $C_x/L_x$ in $JC_L(s)$. Examples are: Case Crunch[7], Lex Machine[8], eLegalls[9], etc.

Legal analytics facilitate $LA_L(s)$ to perform extensive analyses on various legal documents of distinct nature. Examples are: Gavelytics[10], Fast Case[11], etc.

Legal document discovers enable the detection and discovery of relevant documents during legal process and proceeding and label them suitably to the $C_x$ at hand. Examples are: Exterro[12], Everlaw[13], etc.

Legal document developers offer various types of templates of legal documents to aid $LA_L(s)$ to further designing, drafting and developing final and fair simple, moderate and complex legal documents. Examples are: Rocker Lawyer[14], Legalmation[15], etc.

Legal managers are legal tech graphical dashboards, which offer and enable $LA_L(s)$ to administer numerous business operations. Examples are: Brevia[16], Law Geek[17], United Lex[18], Ravn[19], etc.

Legal auto assisted servers are online platforms, which aid and offer legal advice, and legal resolutions in lawsuits and disputes, especially to destitute, $CP_L(s)$ and $AD_L(s)$. Examples are: Legal Zoom[20], Nolo[21], Modria[22], etc.

---

[7] case-crunch.com (accessed: 20.05.2022)

[8] lexmachina.com (accessed:16.12.2023)

[9] elegalls.com (accessed: 16.12.2023)

[10] gavelytics.com (accessed: 30.06.2024)

[11] fastcase.com (accessed: 12.04.2024)

[12] exterro.com (accessed: 16.11.2023)

[13] everlaw.com (accessed: 16.11.2023)

[14] rocketlawyer.com (accessed: 30.06.2024)

[15] legalmation.com (accessed: 16.11.2023)

[16] ebrevia.com (17.01.2024)

[17] lawgeex.com (16.11.2023)

[18] unitedlex.com (accessed: 25.12.2023)

[19] imanage.com/product/artificial-intelligence (accessed: 25.12.2023)

[20] legalzoom.com(18.09.2023)

[21] nolo.com(accessed: 18.09.2023)

[22] tylertech.com/products/Modria (accessed: 16.11.2023)

Majority of the above legal tech examples are based on IaaS, PaaS or SaaS paradigm of cloud computing technology and serve their users in flexible and affordable pay-per-service mode. Although, the classification above is backed up by a smaller set of legal tech examples for each class, but some of them are likely to cut across multiple categories.

Similarly, our *eLegalls* system is also based on SaaS cloud model and is proposed to serve the end users as *eLegalls-as-as-Service*. The fully developed and completely operational *eLegalls* will also intersect through multiple classes — *Legal matcher* and *Legal managers*. Presently, global legal tech landscape is yet sparsely developed and most of these tech tools are still in their nascent stages. However, with rapid upward trajectory of LI and LI-research, faster growth of LI-supported legal tech tools is also inexorable, which will potentially restructure, reshape and transform the current jurisprudence into a faster, fair, and cost-effective civil justice system.

## Conclusion

IT support into jurisprudence, globally, is still in nascent stage and legal world has not yet accessed and exhausted the full potential of IT and its allied technologies; they are: AI, ML, DL, NLP, etc. Particularly, in underdeveloped nations, IT-driven legal ecosystem is apparently either remote or rather imaginary as there are not even thinner traces of LI-centered communications exist yet in several such nations. In developing nations with even enough digital progress and infrastructure, the enthusiasm and manifestation for serious efforts to reshape, repair and transform the senile, lax, and leaden legal system into a LI-enabled faster and fair jurisprudence are yet to be seen. In $JC_L(s)$, legal procedures and practices largely are continued to be conducted and executed in archaic manner. $LA_L(s)$, who are important and inseparable constituents of $JC_L(s)$ still manage and administer their lawyering activities in mediaeval manual fashion, which when viewed from modern digital tech era, cannot be considered -time and effort- efficient. As LI technology evolves, $LA_L(s)$ should also explore new methods and mechanisms.

In that paper we designed, developed and implemented a cloud-based legal tech solution, *eLegalls,* for $LA_L(s)$ as an alternative to their current burdensome and cumbersome lawyering in digital age. *eLegalls* is to aid efficient handling of lengthy legal documents and legal contracts and better managements of legal $C_x(s)$. *eLegalls*, that proposedly serves as a SaaS-

based cloud technology, *eLegalls-as-a-Service*, potentially replaces most manual efforts and $C_x(s)$ management activities of $LA_L(s)$ and equips them with appropriate technology and assists with technology-drive lawyering. Future work includes the expansion of *eLegalls* to further enrich it with more in-need features and functions with the goal to completely eliminate manual efforts and needs of $LA_L(s)$ in handling and managing their $C_x(s)$. Eventually, *eLegalls* will be shaped into an efficient, effective, and robust LI-enabled cloud-based legal tech, which serves and contributes towards emancipating $JC_L(s)$ from archaic, outmoded, and obsolete practices.

# References

1. Alam T. (2021) Cloud Computing and its role in the Information Technology. *IAIC Transactions on Sustainable Digital Innovation,* vol. 1, pp. 108–115.

2. Araujo R.F. (2013) Getting Started with Eclipse Juno. Mumbai: Packt Publishing, 256 pp.

3. Black N. (2013) Lawyers, Cloud Computing, and Innovation: How Cloud Computing Facilities Innovation in the Delivery of Legal Services. *ISJLP*, no. 9, p. 593.

4. Corrales M., Fenwick M., Haapio H. (eds.) (2019) *Legal Tech, Smart Contracts and Blockchain*. Singapore: Springer, 276 p.

5. Dykstra J., Riehl D. (2012) Forensic collection of electronic evidence from infrastructure-as-a-service cloud computing. *Richmond Journal of Law & Technology*, vol. 19, p. 1.

6. Engstrom D. F., Gelbach J. B. (2020) Legal Tech, Civil Procedure, and the Future of Adversarialism. *University of Pennsylvania Law Review*, vol. 169, p. 1001.

7. Fitch A. (2020) Would You Trust A Lawyer Bot with Your Legal Needs? *Wall Street Journal*. 10 August.

8. Geewax J.J.J. (2018) Google Cloud Platform in Action. N. Y.: *Simon and Schuster, 632 pp.*

9. Goswami J.J. (2018) Legal Informatics. Doctoral dissertation. Dhirubhai Ambani Institute of Information and Communication Technology.

10. Guérin S., Polet G., Silva C. et al. (2021) PAMELA: an annotation-based Java Modeling Framework. *Science of Computer Programming*, p. 102668.

11. Hickson I., Berjon R., Faulkner S. et al. (2016) HTML5. A vocabulary and associated APIs for HTML and XHTML. *W3C Recommendation*

12. Lettieri N., Altamura A., Faggiano A., Malandrino D. (2016) A computational approach for the experimental study of EU case law: analysis and implementation. *Social Network Analysis and Mining*, vol. 6, no. 1, p. 56.

13. Lettieri N., Altamura A., Malandrino D. (2017a) The legal macroscope: Experimenting with visual legal analytics. *Information Visualization*, vol. 16, no. 4, pp. 332–345.

14. Lettieri N., Malandrino D. (2018) Cartographies of the legal world. Rise and challenges of visual legal analytics. In: 22nd International Conference Information Visualization, pp. 241–246. IEEE.

15. Lettieri N., Malandrino D., Vicidomini L. (2017b) By investigation, I mean computation. *Trends in Organized Crime*, vol. 20, no. 1–2, pp. 31–54.

16. Marchant G., Covey J. (2018) Robo-Lawyers. *Litigation Journal*.

17. Markou C. (2017) Are We Ready for Robot Judges? *Discover Magazine*. Newsletter, pp. 1–3.

18. Markovic M. (2019) Rise of the Robot Lawyers. *Arizona Law Review*, vol. 61, p. 325.

19. Morningstar Law Group (2015) The Laws of Cloud Computing: Weathering the Storms of Cyber piracy, Hacking, and IP Infringement. Available at: https://morningstarlawgroup.com/insights/cloud-computing-legal-issues/ (accessed: 16.04.2023)

20. Mulligan S.P. (2018) Cross-border data sharing under the CLOUD Act. *Congressional Research Service*, pp. 1–28.

21. Murley D. (2009) Law libraries in the cloud. Retrieved from Law library Journal, vol. 101, no. 2. Available at: http://www. aallnet. org/main-men/publications/ilj/LLJArchives/Vol1-101/pub_iij_v10l (accessed: 20.11.2023)

22. Ngaogate W. (2020) Integrating Flyweight Design Pattern and MVC in Development of Web Application. In: Proceedings of 2nd International Conference on Information Technology and Computer Communications, pp. 27–31.

23. Oh G., Kim S. et al. (2015) SQLite optimization with phase change memory for mobile applications. *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1454–1465.

24. Python J., Go P. H. P. (2015) Google App Engine. *Development*, vol. 1, p. 35.

25. Reingold B., Mrazik R., D'Jaen M. (2010) Cloud Computing: Whose Law Governs the Cloud? Part III. *Legal Works, West Law*, pp. 1–6.

26. Rossi A., Ducato R. et al. (2019) Legal Design Patterns: Towards A New Language for Legal Information Design. In: Internet of Things. Proceedings of the 22nd International Legal Informatics Symposium IRIS. Weblaw, pp. 517–526.

27. Ruhl J.B. et al. (2017) Harnessing legal complexity. *Science*, no. 355 (6332), pp. 1377–1378.

28. Sharma S. (2015) Evolution of as-a-Service Era in Cloud. *Ar Xiv preprint arXiv*:1507.00939 (accessed: 25.04.2022)

29. Sharma S., Gamoura S., Prasad M. D., Aneja A. (2021) Emerging Legal Informatics towards Legal Innovation: Current status and future challenges and opportunities. *Legal Information Management,* vol. 21, no. 3–4, pp. 218–235.

30. Sharma S., AL R.S. (2021) ELegalls: Enriching a legal justice system in the emerging legal informatics and legal tech era. *International Journal of Legal Information*, vol. 49, no. 1, pp. 16––31.

31. Varghese B., Buyya R. (2018) Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, no. 79, pp. 849–861.

32. Vazquez A. (2019) Integrating LDAP with Active Directory and Kerberos. In: Practical LPIC-3 300. Berkeley (Cal.): Apress, pp. 123–155.

33. Wittig M., Wittig A. (2018) Amazon web services in action. N.Y.: Simon and Schuster, 552 pp.

34. Volokh E. (2019) Chief Justice Robots. *Duke Law Journal,* vol. 68, p. 1135.

**Information about the authors:**

S. Sharma — PhD, Founder/CTO.
R. Shandilya — PhD, Associate Professor.
D. Dwivedi — LLB, Advocate.
M. Pant — PhD, Professor.