ADOPTION OF OPEN SOURCE SOFTWARE IN INDIAN BANKING AND FINANCIAL SECTOR

WHITEPAPER ON

Institute for Development and Research in Banking Technology
(Established by Reserve Bank of India)
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>01</td>
</tr>
<tr>
<td>1: Open Source Software: An Introduction</td>
<td>02</td>
</tr>
<tr>
<td>1.1 Reasons to Adopt OSS</td>
<td>03</td>
</tr>
<tr>
<td>1.2 Factors Affecting the Adoption</td>
<td>04</td>
</tr>
<tr>
<td>1.3 OSS Licensing</td>
<td>06</td>
</tr>
<tr>
<td>1.4 Real Time Survey with Indian Banks</td>
<td>07</td>
</tr>
<tr>
<td>2: Use Cases of Open Source</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Banking Workflow Tools and Technologies</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Productivity, Analytics and MIS</td>
<td>13</td>
</tr>
<tr>
<td>2.3 HR and IT Project management</td>
<td>18</td>
</tr>
<tr>
<td>3: Risks, Challenges and Best Practices</td>
<td>21</td>
</tr>
<tr>
<td>3.1 Challenges</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Best Practices</td>
<td>23</td>
</tr>
<tr>
<td>4: Roadmap</td>
<td>25</td>
</tr>
<tr>
<td>4.1 IOSSB: IDRBT OSS Support System for Banks</td>
<td>25</td>
</tr>
<tr>
<td>Appendix: Comprehensive list of Open source Alternatives</td>
<td>27</td>
</tr>
<tr>
<td>Glossary</td>
<td>28</td>
</tr>
<tr>
<td>References</td>
<td>29</td>
</tr>
<tr>
<td>Contributors</td>
<td>30</td>
</tr>
</tbody>
</table>
FOREWORD

“ANY sufficiently advanced technology is indistinguishable from magic” is one of science fiction writer Arthur C. Clarke’s famous laws. Technology landmarks that have dramatically altered human lifestyle include the wheel, the use of Zero, the printing press, the steam engine, electricity, transistor, radio, television and the computer and Internet.

The unbelievable pace at which the last two - Computers and Internet - are evolving can be analysed from two perspectives. The brawn - Hardware (computing power, storage and network) - is nowadays readily available as an affordable commodity on demand. The brain- Software (system, applications, middle-ware) - is evolving more slowly due to its inherent complexity, cost and non-conformance to open interoperable standards.

Open Source Software (OSS) paradigm is a promising approach for collaborative efforts by the best minds to address this challenge. In its “bazaar”-like approach, the entire life-cycle - from ideation to design to development to support and documentation - is done in the open. Public repositories host the source code, which is freely available for anyone to study, improve and add more features. Building on the common base, this leads to very fast development cycles and also to very quick discovery and fixing of any bugs or security flaws.

Linux, Apache, Postgres, Hadoop, Docker and Kubernetes is a tiny sample of major OSS projects that have contributed significantly to the greater common good.

While many resources and studies are readily available on the OSS paradigm in general and its impact on Software development, this “Whitepaper on Adoption of Open Source Software in Indian Banking and Financial Sector” prepared by IDRBT with contributions from several leading Banking and Financial Sector institutions meets a very important need. It not only studies the potential and benefits of OSS for Indian Banking and Financial Sector, but also documents in reasonable detail several inspiring success stories achieved in the last five years.

The learning from these efforts will be invaluable for other institutions as they embark on their respective exciting journeys to harness the best of OSS to deliver value for their organisation. In this context, IDRBT’s plan to form IDRBT OSS Support System for Banks (IOSSB) described in the last chapter should give much needed confidence and lead to the development of a stable, reliable and invaluable eco-system supporting the use of OSS in India.

It is my privilege and pleasure to recommend this well-crafted whitepaper. Widespread adoption of OSS is an essential step in our journey towards Atmanirbhar Bharat, and this whitepaper is an ideal place for Banking and Financial Sector institutions to start.
Executive Summary

The impact of the Open Source Software (OSS) industry is very broad to banks. According to Gartner, 95% of mainstream IT organisations leverage open source software assets within mission-critical IT portfolios directly or indirectly (for example, indirectly through commercial proprietary software using OSS libraries). The Indian government has also been promoting the use of open source technologies in the e-Governance domain.

In the past few years, owing to a rapidly changing business scenario, banks are remodeling their IT divisions, adopting new technologies and methodologies like Cloud, Micro-services, Open APIs and, Open Source (often these different adoptions enforce each other). Banks have started to evaluate OSS adoption efforts with TCO and risk assessments for production scenarios that fall within an acceptable risk-reward threshold.

"Open source" is a model for the development, support and distribution of software that encourages community stewardship of the technology. In some areas, open source solutions rank as industry-leading products such as LINUX operating system, open source RDBMS, programming languages and DevOps automation tools. Few examples of leading innovation platforms being used in banking and financial service organisations are Big data - Apache Hadoop, Apache Cassandra, MongoDB, Cloud - OpenStack, Red Hat OpenShift, Cloud Foundry, Docker, Web development - AngularJS, Node.js, React, Machine learning – TensorFlow and other ML libraries. The recent acquisitions of open source companies by large established corporate tech-vendors is the direct proof that the open source movement has achieved a certain level of maturity.

Even as OSS permeates the portfolios of nearly every mainstream technology organisation (banks), it is not replacing COTS applications outright. Factors, such as programming skills, bandwidth, and self-support options limit the adoption of OSS for mission-critical applications. In the current scenario, OSS solutions are either supplanting and replacing many closed source technologies in a select few areas (e.g., Development tools, machine learning, web technologies etc.) or OSS solutions are augmenting and extending closed-source investments (e.g., middleware, databases). With respect to enterprise end-user applications in Banking, which demands extended support and stability, the lack of compatibility and switching costs alone dictates that OSS solutions will remain few years behind. Consequently, the use of OSS by banks will require a third-party contracted service and support for sufficient technical and legal assurances.

This whitepaper highlights the success stories on open source implementation in the Indian Banking and Financial Sector during the last five years. It also discusses fundamentals of open source technologies, several use-cases of OSS application in Indian Banking and Financial Sector, associated risk and challenges, best practices, and IDRBT’s roadmap on the framework of OSS support system for Banks.
1. Open Source Software: An Introduction

Open source software is a computer software that allows users to modify the source code along with the permit. Public sharing of source code software makes the community of open source more active and vibrant in the software community. Commercial software packages provide users with limited access. The free trial period provides users with restricted features. The desired customer can observe and use some of the primary characteristics during this trial period. Most such software packages don’t allow the modification of the application source code. This makes open source software modules in the corporate world more acceptable.

Establishing Internet technology has marked a fast revolution in open source operating systems history. During this moment, various Linux kernel flavors arose. Concurrent Versions Systems (CVS), code repositories (GitHub), etc., reinforced the community of operating system development to systematically develop and share source code for different apps. Technology giants such as Red Hat, etc., provide adequate financing for the creation of open source software. The economic assistance helps the developer community to communicate with each other through worldwide conferences, Free and Open Source Software (FOSS) meetings, etc.

Apart from the availability of free open source software along with the license to modify and use, the advantages of open source software include greater opportunities for software application architecture and ease in managing and understanding for the developers. The Indian government has also been promoting the use of open source technologies in the e-Governance domain within the country. In this context, the Ministry of Electronics and Information Technology (MeitY), Government of India, has formulated the “Policy on Adoption of Open Source Software for Government of India” to encourage the formal adoption and use of Open Source Software (OSS) in Government Organisations. The National Policy on Information Technology 2012 mentioned the following implementation mechanisms [1]:

* All future Requests for Proposals (RFPs) shall include a mandatory clause for considering Open Source Software (OSS) as a preferred option in comparison to Closed Source Software (Proprietary Software)
* Government Organisations shall ensure compliance with this requirement and decide by comparing both OSS and CSS options with respect to capability, strategic control, scalability, security, life-time costs and support requirements
* GoI shall establish suitable support mechanism for the available OSS that includes Institutional Mechanism, Partnership with Industry, Academia and OSS Community
* GoI shall actively collaborate with OSS communities in India as well as at the International level and contribute wherever in India. The policy coincides with another broad initiative, Digital India.

A successful implementation of OSS consists of different levels of development, which make them suitable for building minor, and comprehensive real-world system. The technologies needed for the functional areas of banking are large, e.g., CRM and Reporting Tools, Project Management/Change management tools, Networking and Monitoring tools, anti-money laundering compliance, assets and liability management tools, learning management system and many more.
1.1 Reasons to Adopt OSS

Cost-effectiveness

- Lower software costs - no licensing fees
- Lesser hardware cost – usually solutions are easily portable and compressed
- Avoids functionality overloading caused by bundled proprietary solutions.

Flexibility and Agility

- Organisations set flexibility to use the platform according to the requirements
- Better reusability of code - open source software gets adapted to a variety of use cases
- Agility ensures that the organisation is keeping up with the competition – OSS enables agility by offering multiple solutions to a problem - helps IT team when it gets blocked due to a particular capability.

Speed

- OSS provides ability to start quickly - Take the community versions, get started, understand whether they can solve your business problem, and if it solves, then begin to deliver value right away
- Start small and quickly with community versions and then scale up.

Reliability

- Because of more eyes to inspect, the reliability tends to be superior as well - The output tends to be robust, tried, and tested code [9]
- Even under strict conditions, various programs such as HTML, Perl, HTML and Apache have been proven to be robust and reliable.

Quality

- Developed by countless users, the platform consistently improves the quality of the product
- New and innovative features get added and the product gets enhanced.
- A major reason to get adopted.
- No commercial pressure that often degrades the quality of the software
- The power of the crowd - Many hands can deliver powerful outcomes

Support Options

- The global communities unite around improving the solutions and troubleshooting more effectively than internal teams working on proprietary solutions
- Paid support options available. In most of the cases, the price lies far below the one the proprietary vendors usually charge
- Most Open Source companies also provide maintenance and support.

Transparency

- Full visibility into the code base, as well as all discussions about how the community develops features and addresses bugs [9]
- No vendor lock-in risks
- Should be careful to use true open source solutions, rather than those from providers that repackage open source software to include proprietary hooks. [9]

Enhances Security

- Much more thoroughly reviewed and vetted by the community
- Quick responsiveness of the open source community and vendors might help in solving information security problems. But make no mistake, simply being open source is no guarantee of security - implementation of open source technologies should be taken up in compliance with the Information Security (IS) policy of the Bank.
For proprietary software, handling security vulnerability is much more complex and time taking.

- With OSS, since you have access to the source code, you can often apply fixes with the help of community, both large and small, at your own convenience, not at the convenience of the publishing organisation's release cycle.

**Scaling**

- With varied options for clustering, load balancing and open source applications, organisations can easily scale up and achieve higher growth.
- Plugins can be easily added to open source programs.

**Merit-based**

- The main objective of open source coding, is to make the most useful product possible.
- Commercial application - Choices of making own decisions in development and customisation.
  - Users or organisations can customise the code for their use cases.

**New Trend**

- Many large enterprises implementing open source solutions.
- Microsoft now counts itself as the world's biggest open source contributor, with IBM and SAP also breaking into the top ten, and in 2018 paid $7.5 billion to acquire one of the original exponents for open source software, GitHub, and its vast code repository [10].

- Fortune 500 companies allow open source to drive their organisations by encouraging developers to use OSS to improve software packages constantly [11].
- Linux is one of the largest OSS example as 94% of the world's top 500 super-computers runs on Linux [12].
- Open source web server Apache has grabbed ~65% web server market share [12].
- Gartner gives high marks to open source vendors in its new Magic Quadrant for Advanced Analytics [12].

**1.2 Factors Affecting the Adoption**

One cannot deny the increased adoption of OSS worldwide. In this context, MeitY has issued the “Framework for Adoption of Open Source Software in e-Governance Systems” in 2015 [2], which discusses different identified factors that influence the adoption (shown in figure 1.1). In the figure, the green text shows the positive influence, whereas the text in red colour show the negative influence. There are a few criteria that can be specifically considered to be of high importance in the Banking and Financial Sector, such as the security and openness (shown in blue coloured text). Similarly, some of the items – can be both positive and negative. E.g. Outsourcing impact – easier to get people trained on common OSS solutions (shown in blue colored text).
Figure 1.1: The elements impacting the selection of OSS [2]
1.3 OSS Licensing

Software solutions are deployed under different types of licenses and categorised as:

- **Proprietary Software** - Access and modification of the source code (whole or part) is prohibited - revenue and profit drive the development
- **Shareware** - Executables of the software are made available for restrictive-use free of charge for a specific trial-period
- **Freeware** - Executables of the software are made available for restrictive-use free of charge permanently
- **Open Source Software** - Source code is available to study and modify the software and to redistribute copies of either the original or modified software.

Free software respects certain freedoms on the part of users, which proprietary software does not respect. According to Richard Matthew Stallman, (an American free software movement activist and programmer) - "the free software movement campaigns for freedom for the users of computing; it is a movement for freedom and justice. By contrast, the open source idea values mainly practical advantage and does not campaign for principles". Free Software Foundation (FSF) [13], founded by Richard Stallman in 1985, is a non-profit with a worldwide mission to promote computer user freedom and sponsors the GNU Project.

The fundamental purpose of open source licensing is to deny anybody the right to exclusively exploit a work [4]. Open source licensing makes possible three substantial improvements over traditional proprietary commercial software licensing models - Innovation, Reliability, and Longevity [3]. The basic principles of OSS licensing (as per the Open Source Definition [14]) are as follows,

- Free Distribution
- Source Code

- Derived Works
- Integrity of The Author’s Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavor
- Distribution of License
- Must Not Be Specific to a Product
- Must Not Restrict Other Software
- Must be technology-neutral

Open source Licenses fall under two categories [13]:

- Those with the aim to have minimal requirements about how the software can be redistributed (permissive licenses) e.g. BSD license and the MIT license
- Those that aim to preserve the freedoms that is given to the users by ensuring that all subsequent users receive those rights (copyleft licenses). E.g. GNU General Public License (GPL). It states: Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Hybrid licensing models – such as dual licensing - are in current trend. In a dual licensing model, the vendor develops software and offers it under an open source license but also under separate proprietary license terms. With respect to the rights provided, multiple versions of software may belong to the same or different class of licensing. Figure 1.2 presents the multiple level of rights provided to different class of licenses [15].

There are 100+ open source licenses available. These licenses can be compared to each other based on:

1. The type of the code linkage with a different licensed code
2. The type of distribution of the code to third parties
3. The type of modification of the code by a licensee
4. Whether patent is granted or not, or manually granted.
Table 1.2: Rights available with different class of licenses

<table>
<thead>
<tr>
<th>Rights Granted</th>
<th>Public Domain</th>
<th>Permissive FOSS License</th>
<th>Copyleft FOSS License</th>
<th>Proprietary License</th>
<th>Trade Secret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyright Retained</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Very Strict</td>
</tr>
<tr>
<td>Right to Perform</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Right to Display</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Right to Copy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, Under Same Licence</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Right to Modify</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Right to Distribute</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes, Under Same Licence</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 1.2. Rights available with different class of licenses

5. Whether modification to the code must be shared with the community or may be used privately
6. Whether modified code may be licensed under a different license or must retain the same license under which it was provided
7. Whether the Trademark grant is available or not.

For a detailed comparison of open and free software licenses, readers may refer [7].

1.4 Real Time Survey with Indian Banks

A real time survey on “the usages of OSS at different functional areas of banking and financial institutions” was conducted amongst Indian banks using a systematic questionnaire. The questionnaire was categorised into four major sections:

1. General-purpose applications - Desktop software, office-suites, databases, web development and server tools, content management, middle-ware and security related software (encryption, PKI, identification and authentication tools)
2. Domain specific applications - Core Banking System (CBS), HRMS, Internet Banking, Mobile banking, e-Learning, Intranet and other applications.
3. Future plans of OSS adoption
4. Suggestions.

The survey contained check-box enabled questions for which the bank authorities had to give their response based on the level of usage and awareness of a specific OSS. The check-box option was categorised into seven ranges (Unaware, Aware, Interesting, Under evolution/trial, Commitment, Limited Deployment, General Deployment) based on the level of acceptance of a specific OSS. We also tried to find out the factors banks might consider for
adoption. For this purpose, the factors we identified are:

- Case studies and examples of OSS adoption in bank
- Return-On-Investment analyses on using OSS applications
- Evaluations of the reliability and security of OSS applications
- Feature comparison among OSS and proprietary solutions
- Descriptions of OSS licenses and legal constraints on what can be done with OSS

We received valuable responses from 15 banks which not only include large banks but also include small cooperative banks. 67% responded that they are using OSS either in limited or full deployment. OSS software usage was further categorised into general purpose OSS and domain specific OSS. Moreover, among the 67% of the OSS using banks, 54% of the banks use OSS for the general-purpose solutions and 33% of the banks use OSS for both the general purpose and domain specific applications. The statistics of different types of general purpose OSS is shown in Figure 1.3. We can observe that under this type, the highest percentage of open source software is used in desktop utilities which accounts to 22% and in content management banks use the least percentage of open source software. Banks are slowly increasing into OSS usage by repeated training and deployment sessions by various OSS promoting organisations.

The percentage of OSS types used by domain specific category is shown in figure 1.4. These types include OSS usage for biometric, UPI, helpdesk management, database activities, etc.

Awareness programs and OSS workshops conducted by various organisations are considerably helping the future OSS adoption plans of banks. Figure 1.5 shows the percentage of banks responses across a scale of score ranging from 1 to 7, where 1 signifies the least likely future adoption of OSS and 7 signifies the most likely future adoption of OSS in banks. The general deduction from the response of banks is that most of the banking sector moving towards adopting OSS in their daily transactions. Still, the adoption of OSS in banking applications is restricted owing to some rules and regulations.
Various identified categories of rules and criteria are shown in Figure 1.6. Banks accord more priority to maintain reliability and security in their transactions. Legal issues and other licensing activities are also considered equally by banks to adopt OSS.

**Suggestions and Response from Banks**

The last section of the survey collected the general comments and suggestions from the respondents. The summary of it is as follows:

- Banks encourage the idea of more and more usage of open source software. However, the clarity of the difference between Open source (with support), Open source (without support) and freeware is near absent in the banking sector. Banks have suggested IDRBT to work towards a common framework for the usage of Open source and Freeware in banks.

- Another important requirement is a baseline for use of Open Source applications in banks by banking research organisations. They advocated that research organisations like IDRBT should launch a common framework through which the OSS applications related to banking can be promoted and distributed.

- The nature of support facilities for OSS should be made clear. Some of the banks shared their concern about the usage of OSS software without the prior approval of higher authorities.

- Banks acknowledged that the huge installation and maintenance cost is saved by using Libre Office as the main office suite and running OSS applications in Core Banking System, HRMS, Internet Banking and Mobile Banking.

![Figure 1.5: OSS Adoption Score Ratio of banks](image)

![Figure 1.6: Adoption Criteria of OSS in banks](image)
2. Use Cases of Open Source

This section covers the details of OSS deployments across organisations for various solutions. Information from multiple organisations has been merged in a few cases. Many organisations have reported usage of generic tools, platforms, languages and applications like Linux, MySQL, PostgreSQL, Java (OpenJDK), Apache (HTTP server), Python, Perl, etc. Since these are individual components for a bigger solution, they are skipped in this section. Almost all the case studies presented here have been operational for a year or more. All the views/comments on commercial products in this section are from the banks/financial institutions, which contributed to this section and not from IDRBT.

The following broad categories of applications are covered:

1. **Banking Workflow Tools and Technologies**
   - QuickFix for FX Trading
   - Usage of Libre Office for documents

2. **Productivity, Analytics and MIS**
   - Big Data Implementation using a variety of OSS around HDFS
   - Social media analytics including automatic speech recognition (voice in video feeds)
   - Customer selection for cross-selling products using R

3. **HR and IT Project Management**
   - Learning management system using Moodle
   - Project management and control tools – like Redmine, SVN etc.

2.1 **Banking Workflow Tools and Technologies**

**QuickFIX Engine for FX Trading**

This section covers the implementation of the Open Source FIX engine “QuickFIX” for Foreign Exchange Trading.

An existing FX dealing system offers both Order Matching and Negotiation Modes for dealing. The platform covers inter-bank spot, swap and other transactions in USD/INR currency pair. Members (banks and authorised Foreign Exchange dealers) typically connect to the dealing system through a proprietary “Dealer Work Station” application. In order to enable members to use their own trading applications to connect to the FX dealing system, it became necessary to provide a suitable Application Programming Interface (API) to such members, to facilitate systems integration.

FIX (Financial Information Exchange) protocol is a message standard developed to facilitate the electronic exchange of information between trading participants in the capital markets. It was decided to use the FIX protocol for communication between the FX Dealing system (or “host”) and the trading applications of members through FIX APIs. To achieve such a solution, it was necessary to have a FIX gateway server that would effectively manage FIX sessions and communication between the member’s FIX gateways and the host system.

After evaluating different options, an open source solution called QuickFIX was selected as the FIX gateway server for this implementation. This is a Java based solution that can connect to any type of FIX servers of the member banks’ end, to establish a channel of connectivity for the flow of market information, order, and trade requests. The main advantages of “QuickFIX” are:

- Free and open source
- Full source code available for C++, Java and .net versions
- Supports FIX versions 4.0 - 5.0SP2
- Runs on any hardware and operating system
- Support for protocol customisations

For seamless communication amongst participants, the solution needs to ensure that the message
format is common. Currently, there are a few banks who are on this platform and are using their internal trading applications to seamlessly connect to the FX Dealing System to get market information, to place orders and to trade in the USD/INR segment.

Journey so far

Following the implementation of the FIX gateway, there has been significant growth in order-flow into the FX Dealing System and a substantial percentage of trade value, driven by members connecting via the FIX API. The implementation has been relatively stable, with no issues attributed to the core codebase of the Quickfix solution.

The next step in the journey is to extend the use of FIX based APIs to other members who stand to gain efficiency through a seamless interface with the FX Dealing System.

Office Suite of Apps – Libre Office

The Bank (TB) is extensively using Libre Office instead of commercially available Office Suite product for day to day operations. Bank had around 18,500 desktops but procured only 4,500 commercial product licenses. Earlier, it was receiving many requests for new commercial product licenses for branches and the cost for procuring each license was quite exorbitant. Considering this TB's IT team suggested branches to start using Libre Office. Initially, there was some resistance from users informing that it is slow and file compatibility issues with earlier commercial product manufacturing company. TB analysed the issues raised and the following were the observations:

1. As TB has provided the latest version of Libre Office it was slow in opening up (high application launch-time). Some of the users of the TB were using older versions of the commercial product, which used to open up fast. The time taken for Libre Office to open up was almost the same as that of the new commercial product version.

2. The second problem was users were used to saving files in .odt default format in Libre instead of .doc format and .odt format was not compatible with the commercial product.

From Head Office, the IT team gave suitable advisories about the above issues and subsequently the staff started using Libre Office without any issues. Now TB has started installing Libre Office in all new PCs being delivered at branches/offices. With the use of Libre Office instead of the commercial product, TB has saved approx. Rs. 25.00 crores one-time cost and Rs. 5.00 crores per annum recurring cost.

Real Time Channel Monitoring System (RTCMS)

The need for Business insights into real time data about customers and partners and the single Centralised Monitoring for Business, Log, System, Database and Applications for Mobile Banking was the real need at an existing Banking Organisation (BO). With digital channels such as Mobile Banking gaining momentum in terms of customer base, transaction volume and partner ecosystem, BO is overwhelmed in her endeavour to keep this dearest yet youngest sibling of all the channels healthy and available. With the major shift in consumer behaviour to use Mobile as a de-facto channel for banking, the executives and digital heads at BO were expecting to have a laser focused view on the 'Overall Performance' of the Channel. BO wanted to commission an Observatory system to have a 360-degree view of the Channel, 24X7 and real time alerting capability.

BO identified Real Time Channel Monitoring System (RTCMS) as a solution. Most of the Monitoring System will focus on Log or Network or System or Applications. However, RTCMS is a single enterprise solution primarily intended to provide a real-time summary of business activities to COO, CEO, CIO, operations managers and Sr. Management. It's also
a single solution used by the IT team to monitor System, Application and Log.

**BO Key Stakeholders**

- Banks Leadership Team – COO, CIO, and CEO for monitoring the Business in real time.
- Business Functional Product Team: for monitoring the performance in real time.
- Support Team: For proactive customer Support
- Infrastructure: To monitor the Operating System, Database and Applications
- Risk Team: Log Filtering, Geo Location and SOC Integration
- Developers: Centralised log for root cause Analysis

**BO Dashboards Configured**

The most visible feature of RTCMS is presentation of information on dashboards processed in REAL TIME / NEAR REAL TIME containing the key performance indicators (KPIs) used to provide assurance and visibility of activity and performance.

**Business Dashboard**

- Being able to monitor the growth of the Channel in terms of No. of subscribers
- Ability to monitor the transaction growth - Volumes and Revenue
- Being able to monitor the Fund Transfer
- Monitor the partner channel availability and latency such as Bharat Bill Pay, etc.
- Business Activity Monitoring – Fund transfer and 3rd Party Products
- Business Trend Analysis - Based on Moving Average

**IT Dashboard**

- To monitor the overall system performance
- To monitor the Mobile Server, Database and Server Nodes.
- A consolidated Application Log Analyser for Developers to quickly troubleshoot issues

- Dashboard from Infrastructure point of view - Systems Dashboard, Application Server Dashboard, File System Dashboard, API Latency

**Alerts**

- Notify Product Owners on business level failures such as latency in a Partner system response
- Proactively inform customers on their transaction status
- Notify IT teams on IT level failures such as CPU overshoot, Server crash, etc.

**Major Functionalities covered by Solution**

1. The goals of business activity monitoring are to provide real time information about the status and results of various operations and transactions. The main benefits are to enable an enterprise to make better informed business decisions, quickly address problem areas, and re-position organisations to take full advantage of emerging opportunities.

2. Log Monitoring: This tool will integrate all the logs from different servers in an indexed time series database

3. OS Monitoring: File System, CPU, Memory

4. Database: CPU, Memory

5. Application Server: Thread and Heap Monitoring

6. Application: API latency

7. BAM solutions additionally can send a notification to whole groups of people through e-mails, text messages, according to the nature of the problem.

Specifically, the proposed system at BO is expected to address the needs of following Personas.

<table>
<thead>
<tr>
<th>Executive</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executives</td>
<td>Responsible for Overall Business</td>
</tr>
<tr>
<td>Product Team</td>
<td>Responsible for Monitoring the Business</td>
</tr>
<tr>
<td>Support Team</td>
<td>Responsible for providing Customer Support</td>
</tr>
</tbody>
</table>
IT Team: Responsible for Operating System, Database and Applications
Risk Team: Log Filtering, Geo Location and SOC Integration
Developers: Responsible for troubleshooting & RCA

About Solution

The Digital Team at BO has envisaged the RTCMS to be more than just being a traditional Monitoring system. The team in fact wanted the system to be capability-centric rather than tool-centric. The system is expected to provide a real-time Persona-based Dashboard reflecting the Business & underlying IT performance of the Channel. The system is also expected to layer the business metrics with IT metrics to highlight the deep relationship that these 2 layers maintain.

The Digital Team wanted the system to help catch bottlenecks in both 'Business' & 'Technical' areas, which could potentially erode the channel performance. Moreover, the RTCMS is expected to act as a catalyst and offer a quick turnaround time to fix the fatal failures in the system by alerting the right people at the right time.

The traditional Monitoring solutions are capable of monitoring the Hardware, Network, Servers and Application Metrics. However, these capabilities are not adequate to meet the stated objectives of RTCMS. Hence, the team at BO decided to embrace Open Source Software (OSS) to build the solution. After carefully evaluating multiple tools, the team has found Elasticsearch, Logstash, Kibana (ELK) stack to be the most suitable technical stack to meet the system’s functionality.

ElasticSearch: Is a No-SQL distributed database that can store millions of records and offers a sub-second Query response time. The RTCMS holds all the business transaction related data, monitoring & performance data and log data in this database.

Logstash: The Logstash tracks down the log files for new log entries, Gathers them, Transforms them and Stores them into Elasticsearch. The Logstash is a key solution component that transforms the raw log & db data into a format that is suitable for data analysis.

Kibana: Kibana is a tool that helps build Dashboards. Kibana integrates with Elasticsearch for the Data. The solution uses Kibana to build persona based Dashboards. These Dashboards are available through a Web interface and can be accessed through a URL.

Conclusion

Monitoring a Digital Channel needs a new definition. Being able to layer the monitoring requirements into Business, IT, Partner is the true essence of Digitalisation. Also, in a fast paced world, being on toes and quickly reacting to emerging user expectations is the key characteristic of Digital Channel. This demand for an observatory system that can alert stakeholders in real time is a key requirement. The team at the BO has redefined the Monitoring and created a Model that tracks the Business and IT performance by leveraging the Open Source Software (OSS) - a model that is truly Novel and widely Replicable.

2.2 Productivity, Analytics and MIS

Big Data Analysis Framework

At the start of the Journey

Data at TE continues to grow at an exponential rate every day. The current growth rate is 400 GB per day (100 TB per year) and is expected to grow many-fold as TE starts new lines of business, and as more sophisticated algorithm-based trading methods are deployed. TE has observed that the data volume is doubling every two years.

As per Statuary & Legal compliance mandates, most of the data had to be made available for a number of years to provide back information to authorities on
an ad-hoc and need basis. Its business requires complex on-demand analyses & reporting on terabytes of data. This is coupled with real-time surveillance and fraud detection models, which was challenging with the legacy technologies.

Big Data was the only answer and solution to all needs, that:

- Facilitate an integrated and in-depth view of the enterprise;
- Is a single place where historical and current information is easily available for decision making
- Is faster and flexible data analysis
- Provides consistency of information

Thus, TE developed Big Data Lake on the Hadoop platform.

**Implementation of Big Data**

In the year 2013, strategically TE started its journey in moving towards the deployment of the Next-Generation Analytics Platform on Open Source Big Data Technologies. A few of the important underlying considerations were:

- Consolidation of information from multiple data sources into a scalable and robust data repository via an automated process
- Data-Driven Enterprise: Ability to analyse and act on information in real-time
- Highly Scalable, Less in Total Cost of ownership, High in performance
- Supports future technologies such as Artificial Intelligence & Machine Learning.

TE's Big Data and analytics platform deployment is based on HADOOP ecosystem framework. Though the system has data replication inbuilt into the HADOOP architecture, as per the best BCP practice, TE needs to deploy its Big Data system in a different seismic zone. The objective of the deployment is to make the system available in case of non-availability of the primary site. Hence, TE deployed both Production and Disaster site Big Data lake at different seismic zones.

The main aspect of the solution consisted of a 500 terabyte Big Data Lake to move TE’s traditional information management platform to an open source one.

- Create an Enterprise Data Lake and consolidate TE’s data silos
- Offload legacy appliance-based 15 TB Data and build the MIS on this platform. Include Data from source systems in database management systems
- Migrate TE's Reporting and Analytics and Dashboards to this framework using open source Big Data technologies
- Use this cluster as an intelligent archive and do away with expensive storage, e.g. emails, documents, backups, archives.

TE deployed 500 TB of Big Data HADOOP lake at its primary location and 250 TB Big Data HADOOP lake at its DR location as per the Business Continuity process. The Big Data HADOOP cluster is deployed using Cloudera Distribution of HADOOP.

Following Use Cases implemented:

- Enterprise Data Warehouse offload from legacy systems
- Social Media Analytics on Textual Data, Fraud Analytics
- Real-time Analytics, Streaming Analytics, Near Real-time Machine learning
- Big Data security and Audit compliance
- Big Data Disaster Site implementation at remote place
- Operational Analytics

Objectives were to have Enterprise Data Hub with structure and unstructured data with scalable, highly available, Lambda architecture with a low cost of total ownership.

Overall, the Big Data lake implementation was broken up into four parts
TE’s Experience

1. Seamless transition to the new data platform
2. Data-driven decisions. Using complex analysis, it is now possible to get deeper insights into business, something that was impossible with the earlier platform. Some of the examples are
   - Market Snapshot storing and analysis: Storing and analysing huge semi-structured voluminous data was made possible by the use of open source technologies. The Big Data solution has made it possible to analyse such data in the exchange.
   - Order Log LTP analysis: Due to huge volumes in order books, many exchanges fail to deliver these complex analysis. TE can now process billions of orders along with millions of trades iterating this data millions of times to generate those reports

Performance

Highly Scalable Environment

1. At least 4x improvement in query performance as compared to the current environment
2. It is now possible to run historical queries spanning many years using very complex algorithms
3. TE is able to service most of the ad-hoc historical requests spanning multiple years on the same day without having to resort to tape backup restores.

Operational Efficiency

With the introduction of algorithm trading in the Indian stock markets, the trade to order ratio has gone up to 1:125. It was next to impossible for the old system to analyse matching order with the trade for longer periods, which requires the system to churn billions of orders with millions of trades.

Major Challenges

Initially, like any new technologies, the maturity level of Bigdata HADOOP framework was low. Since the architecture and technology were new, meeting the business goal and designing a robust architecture framework with the best fit of new tools was a big challenge. Besides this availability of required skill sets was also limited. TE has built a startup culture and adopted agile methodologies for the execution of the project. It appointed a startup company along with its own in house team to overcome all the challenges.

Social Media Analytics – Voice in Video Feeds

Building Blocks on Big Data

Social Media Analytics-Driven by highly competitive market conditions and the need to conduct fraud analysis, the enterprise (TE) established an automated solution for Social Media Analytics on Big Data platform using Text Data mining algorithm. TE has adopted data analytics based systemic solution which relies on artificial intelligence and machine learning techniques to track news related to TE listed companies on news websites and social media like Twitter, Facebook, etc. (Digital Media). TE made use of Big Data technologies, namely SPARK Machine Learning, KAFKA messaging, HDFS
storage, KUDU storage, along with various Machine learning algorithm to implement the solution, without using conventionally licensed software.

TE used Spark streaming for automated content tagging. Feedly, a popular news aggregator, is used to get news articles. These are stored in HDFS, processed and then stored in Hive tables. Python does the content tagging and the final results are in a Hive table. Twitter data is also stored in Hive tables. India is a country of diversification in terms usage of languages. Almost all TV channels publish live comments and they are hundreds in numbers. Most TV channels broadcast various comments and news about TE listed companies on a daily basis. Sometimes these speeches which are broadcasted in news channels contains rumours which may have a negative impact on the stock market.

TE is committed to transparency and does continuous market monitoring to check any adverse impact on the market due to rumours. It's practically impossible for the human being to track almost all TV channels live streaming and detect rumours in near real-time. To identify such rumours about TE listed companies in near real-time by analysing live streaming input from various TV channels published in English and Indian regional languages, TE implemented project automated speech recognition (ASR) with the help of machine learning and artificial intelligence-based system by applying deep learning algorithm for natural language processing. The motivation of this work is to develop an efficient ASR system for a language that most common people speak in India. And make it possible to all other Indian languages without much more handcrafted efforts.

**System Architecture**

![Diagram of System Architecture](image)

**Data Collection**

The solution needs a large volume of transcribed data for making an efficient system using deep learning. TE collected 1000 hours of speech data from a popular business TV Channel News data for training. All speech data are sampled at 16 kHz mono recording.

**Model Building**

The system is trained in GPUs. The system is able to handle variable-length batch with the help of
masking. The input speech data is converted into spectrograms by the help of spectrogram module and it is chopped into frames. This spectrogram input is directly fed to the neural network as inputs.

The network module in the training script has done the major role in training and testing of the system by interacting with the neural network and configuring the parameters used for the network. In this model, one can configure the model name and the number of GPUs used for training. The number of epochs which is needed for training the entire data is also configured. This module is responsible for creating log files for the entire train and test results.

A Deep speech module is included the system to establish the RNN and convolution architecture. It is implemented by using the PyTorch framework. In this module, it is mandatory to define the number of RNN hidden layers used. A mapper function is used to map the input tokens to the dictionary file. An evaluator module is used to calculate the word error rates in the system and is responsible for CTC predictions and to handle the numeric to tokens. The word error is calculated on the basis of the predicted and target inputs given to the system. A predict to token function will take the predictions to a sequence of characters with the help of the mapper. The numeric values obtained from the system is converted to a sequence of a character sequence with the help of this mapper.

Around 1/10th of data is used for testing the entire system in all the cases. The word error rate is calculated at the time of testing to evaluate the performance of the system. To calculate the word error rate, TE used the Levenshtein distance and the implementation is done in the evaluator module.

The entire system is designed using open source technologies. It used Baidu’s Deep Speech-2 framework which is scalable to a large extent. Current deployment is using one GPU based server for speech recognition solution, hence, limiting the processing to a few TV channels. But the architecture is highly scalable and adding more GPU based server it can process more and more TV channels.

TE had trained the system with Hindi, and subsequently, it has been training the system with other regional languages. Pretty soon, its business users will be able to detect rumours in many more Indian regional languages.

Challenges while implementing in TE:
- Noise detection
- Recognition of Indian accent
- Sample Data collection for training purposes.

To overcome all challenges, TE needs a very large amount of data for training purposes. Initially, they trained the module with nearly 100 hours of video. The system automatically learned from the audio using deep learning techniques.

Customer selection for Cross-selling of products using R

Propensity Model to target Insurance Offering

The main objective of the initiative is to improve the productivity of the Bancassurance channel through improved conversion rates, higher premium earnings, low contract cost, higher return on investment by cross-selling of Insurance Products to existing Savings Bank Customers. The model is designed to identify top N customers from existing SB Customers who are likely to buy Insurance Products, thereby enabling the Bank to improve Product per Customer Ratio and customer loyalty.

Technology / Solutions Leveraged

Statistical / Predictive Model is developed using opensource tool R.

The solution involves thorough Statistical study of the 360-degree view of the customer in order to arrive at the propensity of the customer to buy an insurance product. The team analysed more than
100 variables and identified statistically best 10% of variables as final variables. They have trained the Model using statistical methods to learn from the data and produce output as a propensity to buy the Insurance Product. The Product is pitched for customer, who has a higher propensity. Hence, the solution is not based on rules and it is validated statistically.

The final solution is scalable as initially it was implemented for one Circle office jurisdiction as POC. Basing on the results from the POC, it implemented the solution for Bank as a whole. The final output is set of top N leads for each branch and branches can view the leads using an internal portal. Hence, last mile connectivity is also taken care of. Post-implementation of the model, the lead conversion rate for the insurance products at the branch increased considerably.

2.3 HR and IT Project Management

Learning management system using Moodle

While many LMS implementations have utilised Moodle as its underlying open source platform; other alternatives like Chamilo can also be used. Bank-Digital-Campus (BDC) is a learning management system (LMS); a software application or Web-based technology used to plan, implement and assess a specific learning process. Please note that BDC in the following text refers to the LMS deployment being show-cased. Alternate deployments of LMS may restrict it to be Intranet-only, deployed with other authentication schemes (e.g. open to the public for general awareness), hosted on community cloud (e.g. IFTAS/IDRBT), may or may not be integrated to PMS/ERP, etc. The Advanced Distance Learning group, sponsored by the United States Department of Defense, has created a set of specifications called Shareable Content Object Reference Model (SCORM) to encourage the standardisation of learning management systems.

The BDC is deployed in a Cloud environment accessible 24 x 7 for all employees of the bank. Access to the application is managed using Active Directory login. This Learning program enables employees to improve their competencies by undergoing e-Learning courses available in specific domain areas. Learning Points is a mandatory learning program designed and implemented in BDC. All employees of the Bank should mandatorily complete the Learning Points based on the Role they perform in the Bank. The points earned through learning program of BDC is linked to Performance Management System (PMS) of the employee.

History and Content Spread

BDC has been operational for over ten years and hosts more than a hundred Learning courses, for more than 10,000 users. Comprehensive Dashboard is available for the users to access courses and track progress on the bank’s employee-facing website.

Authentication

For effective and secure usage, a mixed approach has been adopted for deploying BDC. Although LDAP is largely implemented with open source solutions and as a result has more flexibility than AD, it generally doesn’t have the same concepts of domains or single sign-on. Thus, BDC has been using the Active Directory Federation Services (ADFS), which is a proprietary Single Sign-On (SSO) solution. As a component of Windows Server operating systems, it provides users with authenticated access to applications that are not capable of using Integrated Windows Authentication (IWA) through Active Directory (AD). Developed to provide flexibility, ADFS gives organisations the ability to control their employees' accounts while simplifying the user experience: employees only need to remember a single set of credentials to access multiple applications through SSO.
The Deployment Architecture

Moodle Web Tier
Moodle Web Tier is composed of multiple EC2 instances that act as Web Servers running Linux OS with Apache and PHP. The EC2 instances are bootstrapped using a pre-configured Amazon Machine Image containing necessary software stack along with Moodle Software.

Moodle Content Repository
Static content and resources including video, audio, images and multimedia objects embedded inside Moodle courses are stored in Amazon S3 Storage Service (registered as a repository in Moodle). Amazon S3 is a highly durable and scalable storage infrastructure that is highly available. Objects stored in Amazon S3 are delivered via Amazon CloudFront that automatically edge-caches these objects and routes requests to the nearest edge-location for speedier delivery.

Moodle Database Tier
Amazon RDS MySQL service is being used to host Moodle database. Amazon RDS is a managed Relational Database Service that automatically patches the database software and backs up database and supports point-in-time recovery. It provides the flexibility to scale the compute resources or storage capacity associated with the Moodle database to achieve high levels of performance. RDS Multi-AZ deployment enables to maintain a stand-by replica of master Moodle database in another availability zone to fall back in case of any failures in the master database. RDS Service automatically syncs the standby copy with the master database.

Mobile Application IOS & Android
With the official mobile app for Moodle, you can:
- Browse the content of your courses, even when offline
- Receive instant notifications of messages and other events
- Quickly find and contact other people in your courses
- Upload images, audio, videos and other files from your mobile device
- Track your progress, mark tasks as complete and browse your learning plans
- Attempt quizzes, post in forums and edit wiki pages
- View your course grades

By integrating Moodle with proprietary software, enterprise-grade features are enabled for non-critical application in a variety of platforms (browser on desktop to a mobile application). Additional benefits of open source can be leveraged if the
learning modules hosted in Moodle are also developed using OSS paradigms, thus bringing down the cost of modules across the banks. E.g. learning course on “vishing” or “Bharat QR” is almost identical across all banks. If these modules are developed and publicly shared using git or similar repositories, other banks can adapt and extend it. Using the original principles of OSS, the source banks also benefit as it gets to re-integrate the extended solution.

**Project Management / Change Management**

**Version Control**

The Organisation (TO) was using one commercial product for all its source code and document maintenance for in-house and core software built by technology partners. Although, the commercial products was a good fit initially, it had several limitations. Some of the problems that emerged were:

- The company stopped support for this application
- The commercial product could support only 4GB of data
- Back up of commercial product was cumbersome

To overcome all these, the company introduced another commercial product, which not only had a very high license cost but also involved a high maintenance cost. It required extra investments like Windows server OS and SQL Server database installation licenses as well. While evaluating, the new commercial product as a replacement for the older one, apart from its high cost, implementation of the system was also a challenge, considering the fact that the new commercial product would not work only with company’s products.

It was then decided that TO would evaluate open source products for its version and document control system. While evaluating such products, TO came across Subversion (SVN), a widely used version control system. SVN being platform-independent and powerful in all aspects of version management and document control, fits in perfectly with the requirements of TO. After a thorough evaluation, SVN was introduced into TO’s ecosystem successfully and has been functioning as per expectations thereafter.

Apart from being a lightweight Version & Document control system, SVN uses an inter-file branching model that helps it to manage parallel developments on particular source code. Each branch can also maintain its own history. The changes can be merged back to the main trunk or between branches. SVN comes with its own limitations such as:

- Merging conflicting files can be a major problem
- Security can be a bit challenging to use depending on which server (svnservice.exe or Apache/WebDAV) you choose to run your repositories.

TO has migrated most of its source code from the commercial product to SVN and is substantially on SVN as of this writing.

**Issue Tracker**

TO was using an in-house built application to manage software change management. This software had overheads of code maintenance even for adding a parameter to workflow and masters. After evaluating various products for change management control system, TO came across Redmine application. Redmine is an open source application, which had fully functional change management workflow wherein there is a lot of ease in configuring the application as per one’s needs. The system is working as per expectations and has blended into TO’s ecosystem. It works on totally open source technologies like Apache Tomcat web server and My SQL as the database.
3. Risks, Challenges and Best Practices

In the previous section, the details of OSS deployments across organisations for various solutions are covered. While implementing open source, there are several challenges that an organisation might face. In this section, we enlist the common challenges faced by organisations while implementing and maintaining OSS. We will also discuss the best practices for the successful adoption of OSS.

3.1 Challenges

**Difficulty of Use**

Some open source applications may be tricky to set up, use and may lack of user-friendly interfaces or features. There are several open source software that solve larger problems, but not much attention is given to its Graphical User Interface making the software annoying to work with, especially for non-technical users, affecting productivity and putting off technical staff from adopting or using the programs with ease [8]. However, this issue cannot be generalised for all open source software. For example, Libre Office, Mozilla Firefox and Android OS are easy to use.

- For using such software, organisations need to dedicate some time to train the team for building a proper GUI and integrating it with the back-end, which may require as much time and money as rewriting the whole software.

**Lack of extensive Tech Support**

Although many open source software user communities are out there and can be very responsive to resolve the issues, however, this dependency does not work every time since there is no accountability for the users in the community to fix the bugs, provide proper training, or respond to the questions and requirements. Adopting an OSS also includes inherent complexity while customising the OSS as per IT policy, guidelines and the integration of different components of a solution.

**Compatibility Issues**

Many types of proprietary hardware need specialised drivers to run open source programs, which are often only available from the equipment manufacturer, adding the cost of the project. Even if an open source driver exists, it may not work with the software as well as the proprietary driver [16].

- When evaluating your options, make sure that you assess its compatibility with your existing systems and equipment.

Version control is another challenge - Continuous development happens in parallel in the OSS by several community developers. Hence, the software version changes very frequently unlike commercial software and creating confusion among the team since they are uncertain which version does what and if it is compatible with other software and platforms. Certain platforms do not have provision to integrate with the open source tools.

- Make sure that you assess the stability of the product.

**Exposure to Vulnerabilities**

Without proper handling, OSS are typically open to vulnerabilities, such as identity theft, virus transfers, and other activities that might irritate open source software users. Also, getting patches for fixing the vulnerabilities every time is difficult [8]. Although the thoroughly reviewed code of an OSS enhances security, a few common security vulnerabilities like Cross Site Scripting, SQL Injection, and Expression Injection, etc. might exist and thus a proper approach to handle this problem is needed.

- Examine the software before using it with someone on your team who knows how to use that OSS, and thus determine the level of risk associated with using it.
Liabilities and Warranties

Conventional open source software licenses may contain only limited warranty and no liability or infringement indemnity protection.

- Although most of the OSS come free, while implementing one has to do a thorough scrutiny both technically and legally of the license agreements to check if there were any bindings other than costs.

Hidden Costs

OSS might look free up-front, however for the longer run it might involve some hidden costs, which might be a burden, if not considered. A number of such hidden costs include: administration, training, setting up (you may require new hardware), installation, importing data, integrating with existing systems, customising the software, maintenance, and support (for third-party support services as well as unexpected internal support).

- Carefully assess the total cost of ownership for any software you intend to use.

Unavailability of Skillset for Development and Maintenance

The architecture and technology could be new, which can make meeting the business goal and designing robust architecture framework with best fit of new tools a big challenge. Insufficient knowledge of In-house staff about OSS and in-house support to fix any vulnerabilities of OSS increases the dependency on IT-vendor. Getting proper IT staff that can maintain the Open Source Software products in a timely manner is difficult given that human resource attrition in this area is high. Institutions also faced problems due to unavailability of a common pool of expertise and lack of a common platform for knowledge sharing on OSS.

- Check if the software has its own website where you can learn more about its uses.

Security related Issues

- Lack of strong security policies and proper scrutiny to allow open source to be used within the organisation
- Lack of certain security measures (such as Anti-Virus, etc.) as desired by the Information Security Policy
- Lack of a robust patch and upgrade repository to streamline fixing issues related to security and other aspects
- The stability of a new version of OSS is unknown thereby making it risky to adopt immediately in production environment
- It’s worth checking that the regular and timely security updates or patches are available for the software and there is enough documentation to support installation, setup and software troubleshooting.

Challenges in obtaining support for OSS

- Mostly small IT-vendors are not able to provide on-time skilled-resources on OSS
- Unavailability of dedicated support and/ or AMC for OSS
- Lack of dedicated support specifically for security and performance related issues
- Unavailability of timely patches and/ or updates
- Difficult to get support for OSS bugs/ defects. For few cases, community help is available but is not well organised
- Lack of a robust patch and upgrade repository
- Few OSS are still supporting older version of algorithms and lower level encryption
- In few cases, only paid version of OSS has functionalities like SSL encryption, User management, multi-user access, integrated management tools, etc.
- In certain cases, dependency on OEM to use latest released version of OSS as part of their bundled offering. For e.g., Apache comes as a
bundled package with Solaris 11.3 operating system, hence it cannot be upgraded till the OEM releases the required version

OSS are generic in nature which may contain multiple default packages that may not be in use but may have a potential to create issue if modified and/or removed.

3.2 Best Practices

The following are some of the steps that the organisation and users are suggested to follow for the successful adoption of OSS:

Establish a Guideline

- OSS is simple to install and use, however should be used in a restricted environment with necessary controls to ensure compliance with the IT and IS Policy
- Based on the market and maturity, an organisation may have varying degrees of risk demand. It is important that an organisation prescribe a policy about the usage of OSS, or the product might be shipped with known vulnerabilities and/or incompatible software licenses
- Create a steering committee which will plan and direct the progress of the implementation of OSS in an organisation
- Plan a roadmap to integrate the OSS system to the existing system. Also have a clear roadmap for future migration of existing systems to OSS
- Select a business area/management system where OSS can be implemented initially and zero in on the technology, which will be best suited for handling the project
- Decide a program structure
- Set-up strategies and clearly defined procedures and policies to standardise methods to implement open source applications in the organisation.

Select an appropriate License Model

As discussed earlier, an OSS can be licensed under a variety of versions. For each version the use, attribution and the distribution rights may vary. Most of the times, the community versions of OSS have limited features when compared to enterprise versions and/or similar commercial products. Selecting an appropriate license model ensures that the proprietary IP is protected.

Bring up a Team for Robust Usage

In addition to dedicated developers, the involvement of security experts as well as software engineering experts and management team is equally important. While developers can utilise or built open source packages and implement certain fixes, this can create a false sense of risk mitigation. Security experts can concentrate on security-related issues and protocols that have to be followed. Have a strong Security team to plan a robust security system to plan, detect and weed out threats arising out of the OSS. Software engineering experts can interact with security experts while designing their solutions to business problems. Also, a good technical and management team can research and work on the project. This coordination will help the organisation to reduce the maintenance cost of the software, and the users will be benefited more.

Institutions can build a start-up culture and adopt agile methodologies for execution of the project. To overcome all the challenges, institutions may engage start-up companies along with their own in-house team.

Running Security Tests

It is surely a challenge when it comes to security of open source applications. The IT team has to put these applications under rigorous security tests such as Application Security tests, Vulnerability Assessment and Penetrative tests to detect security related issues. In case of minor issues, the same could be quickly resolved with proper parameterisation and patch deployments. You may
consider the density of security bugs and historical security of a particular project, which might include the average number of security bugs present in each release. Also, the time gap between the discovery of a vulnerability and the release of its fix might also be considered.

**Patch Management**

A proactive approach is very crucial to design the patch management framework to make the infrastructure robust towards vulnerabilities. Technologies to keep track of location and usage of different components can be considered too for quickly identifying the vulnerable applications in the organisation. Also, it is equally important to involve the security experts before applying the patches.

**Segregated and Controlled Repositories**

Keep a separate trusted repository for security-approved applications and programs. This repository will help the organisation to discriminate the open source software easily. The software which needs more modification and development also can be identified by this categorisation. Also, it will help the users to develop trust in the software and the organisation.

An OSS development practice may contradict with the organisation’s internal policies. A good practice is to have a local cached version of popular repositories, allowing the security team to closely control which components (and hence which vulnerabilities) are being included in the final product. It also ensures that only a single approved version of a component is used, rather than multiple (possibly) vulnerable versions.

**Open source Communities**

Open source communities frequently support OSS by volunteer contributions for enhancements, and bug fixings. A project with few contributors or the long gaps in releases may indicate the weak support. One should keep in mind that although OSS communities are available for any support, it may not meet the required SLAs.
4. Roadmap

The financial services industry is considered to be a leader when it comes to implementing new and innovative technologies into their business plans and products. Since the last decade, many financial institutions have begun to explore and apply open source alternatives because of the many attractive attributes, including affordability, flexibility and the availability of the open source community. Others are on the fence regarding the concept of open source, reasons being security issues and the lack of accountability & support as top concerns. Although choosing open source software is simple, as open source continues to grow in your enterprise, getting support is not so simple, especially when you’re running multiple packages at once. In this section, we discuss IDRBT’s strategy for providing a support system for OSS being deployed/planned to be deployed at different functional areas of banks.

4.1 IOSSB: IDRBT OSS Support System for Banks

Looking at the challenges Indian banks and financial institutions face for the adoption of the open source software, IDRBT’s strategy is to have a dedicated OSS Support System for Banks - a community supported structure, the basic framework of which is shown in figure 4.1. The components of the framework are described as follows:

1. Query Submission Portal: The objective of this unit is to collect queries from banks and financial institutions. The inputs which may be needed are: the name of the institution, subject of the query, broad module, the OSS related to the query and the detailed question. All the queries submitted to this portal will be sent to the IOSSB unit.

2. IOSSB Unit:
   a. Initial Classification: At first, based on the internal structure of the framework and the context, the query is classified into a specific OSS of a module. In the initial phase, this step will be performed manually, which will be automated later.
   b. After understanding the class of the query, it will be sent to the IDRBT’s internal expertise, available support systems (specific to individual OSS), banking community support group, and other generic help groups (such as GitHub society, StackExchange help group, FinTechs etc.) to get resolved.
   c. After receiving an appropriate set of solutions to the query, it will be sent back to the sender. At this phase, proper scrutiny will be applied to the solutions received from one or multiple platforms.

Initial stage

- With an objective to provide support to all the popular open source alternatives for Indian banks and financial institutions, we aim to start with
  - Hadoop distributed systems for big data analytics
  - Open source DBMS - MySQL and PostgreSQL
- In the initial phase, the query can be dropped at dedicated email ID. Later, can extend to another platform for providing clarifications with multiple channels so that the banks and financial institutions can refer and collaborate on it.
- Maintain an active and comprehensive list of open source alternatives, which will be regularly updated on a weekly basis.

The Institute will communicate to the banks once the set-up is ready.
Figure 4.1: Framework of IOSSB: IDRBT OSS Support System for Banks
## Appendix: Comprehensive List of Open Source Alternatives

<table>
<thead>
<tr>
<th>Domain</th>
<th>OSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Resource Planning</td>
<td>Adempiere, OFBiz, Dolibarr, ERPNext, WebERP, Opentaps</td>
</tr>
<tr>
<td>Operating System</td>
<td>GNU/Linux, Android, Ubuntu</td>
</tr>
<tr>
<td>Internet Browser</td>
<td>Mozilla Firefox, Chromium</td>
</tr>
<tr>
<td>Content Management Systems</td>
<td>Joomla, Wordpress, Drupal</td>
</tr>
<tr>
<td>Office Suits</td>
<td>Open Office, Libre Office</td>
</tr>
<tr>
<td>Web Server</td>
<td>Apache, Nginx</td>
</tr>
<tr>
<td>Cloud Infrastructure</td>
<td>OpenStack, Cloud Stack</td>
</tr>
<tr>
<td>Virtualisation</td>
<td>VirtualBox, Xen, LinuxKVM</td>
</tr>
<tr>
<td>Database</td>
<td>MYSQL, PostgreSQL</td>
</tr>
<tr>
<td>Big Data</td>
<td>Hadoop, MongoDB, Hbase, Spark, Redis NoSQL</td>
</tr>
<tr>
<td>Application Servers</td>
<td>Apache Tomcat, WildFly, Nginx</td>
</tr>
<tr>
<td>IDE/Development</td>
<td>Eclipse, PHP, Visual studio community edition</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>Rstudio - for R, Anaconda and Jupyter - for Python</td>
</tr>
<tr>
<td>HTML/CSS Editor</td>
<td>BlueGriffon, SeaMonkey, Aloha</td>
</tr>
<tr>
<td>Graphic Program</td>
<td>GIMP, LibreCAD</td>
</tr>
<tr>
<td>Publishing</td>
<td>Scribus</td>
</tr>
<tr>
<td>Digital Repository</td>
<td>Dspace, Eprint, Koha</td>
</tr>
<tr>
<td>Project Management Software (with Issue tracking system)</td>
<td>Redmine, Apache Bloodhound, OpenProject, Tuleap</td>
</tr>
<tr>
<td>Email and Collaboration Software</td>
<td>Group-Office Community, FusionForge, SOGo, Zentyl, Drupal</td>
</tr>
<tr>
<td>Learning Management Software</td>
<td>Moodle, Atutor, Chamilo, Open edX</td>
</tr>
<tr>
<td>Networking and Application Monitoring Software</td>
<td>Snort, OpenVAS, Pandora FMS</td>
</tr>
<tr>
<td>Technology Stack</td>
<td>ELK, SMACK, MEAN, BCHS, LAMP</td>
</tr>
<tr>
<td>Hypervisor</td>
<td>Linux KVM and Xen</td>
</tr>
<tr>
<td>Security &amp; Forensics</td>
<td>Autopsy, Wireshark (for packet capturing and network forensics), FTK Imager</td>
</tr>
<tr>
<td>Mobile Application Security Testing</td>
<td>Burp Suite (free edition), ADT Bundle, Wireshark, WebGoat 7.1, OWASP ZAP 2.5.0, Firefox Web Developer Tools</td>
</tr>
<tr>
<td>Vulnerability Assessment Penetration Testing</td>
<td>Metasploit Framework, Burp Suite (free edition), OWASP Zed Attack Proxy, SQLmap, NMAP (for network scanning)</td>
</tr>
<tr>
<td>Customer Relationship Management and Reporting</td>
<td>JasperReport, Pentaho</td>
</tr>
<tr>
<td>Container Management</td>
<td>Kubernetes, Docker</td>
</tr>
<tr>
<td>Microfinance</td>
<td>Fineract</td>
</tr>
</tbody>
</table>
### Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F)OSS</td>
<td>(Free and) Open Source Software</td>
</tr>
<tr>
<td>TCO</td>
<td>Total Cost of Ownerships</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf</td>
</tr>
<tr>
<td>RTCMS</td>
<td>Real Time Channel Monitoring System</td>
</tr>
<tr>
<td>CVS</td>
<td>Concurrent Versions Systems</td>
</tr>
<tr>
<td>MeitY</td>
<td>Ministry of Electronics and Information Technology</td>
</tr>
<tr>
<td>RFP</td>
<td>Requests for Proposals</td>
</tr>
<tr>
<td>GoI</td>
<td>Government of India</td>
</tr>
<tr>
<td>FSF</td>
<td>Free Software Foundation</td>
</tr>
<tr>
<td>CBS</td>
<td>Core Banking System</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>HRMS</td>
<td>Human Resource Management System</td>
</tr>
<tr>
<td>BSD</td>
<td>Berkeley Software Distribution</td>
</tr>
<tr>
<td>SVN</td>
<td>Subversion</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicators</td>
</tr>
<tr>
<td>ELK</td>
<td>ElasticSearch, Logstash, Kibana</td>
</tr>
<tr>
<td>RCA</td>
<td>Root Cause Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCP</td>
<td>Business Continuity Planning</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>CTC</td>
<td>Connectionist Temporal Classification</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>SCORM</td>
<td>Shareable Content Object Reference Model</td>
</tr>
<tr>
<td>ASR</td>
<td>Automated Speech Recognition</td>
</tr>
<tr>
<td>PMS</td>
<td>Performance Management System</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>AD</td>
<td>Active Directory</td>
</tr>
<tr>
<td>ADFS</td>
<td>Active Directory Federation Services</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign-On</td>
</tr>
<tr>
<td>IWA</td>
<td>Integrated Windows Authentication</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>IOSSB</td>
<td>IDRBT OSS Support System for Banks</td>
</tr>
<tr>
<td>OEM</td>
<td>Original Equipment Manufacturer</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>AMC</td>
<td>Annual Maintenance Contract</td>
</tr>
</tbody>
</table>
References

[8] https://www.nibusinessinfo.co.uk/content/disadvantages-open-source-software
[16] https://www.nibusinessinfo.co.uk/content/disadvantages-open-source-software
IDRBT Staff Paper Series

Best Practices From IDRBT

AI in Banking
A Primer

HANDBOOK ON APPLICATION PROGRAMMING INTERFACES (APIs)

WHITE PAPER ON 5G APPLICATIONS FOR BANKING AND FINANCIAL SECTOR IN INDIA

BLUEPRINT OF BLOCKCHAIN PLATFORM FOR BANKING SECTOR AND BEYOND

These publications can be downloaded from www.idrbi.ac.in.
These publications can be downloaded from www.idrbt.ac.in.