ID4D
IDENTIFICATION FOR DEVELOPMENT

Open Source for Global Public Goods

WORLD BANK GROUP

1818 H Street, NW, Washington, D.C., 20433
Telephone: 202-473-1000; Internet: www.worldbank.org

Image credit: iStock/Jakarin2521

Contents

About ID4D

The World Bank Group's Identification for Development (ID4D) initiative uses global knowledge and expertise across sectors to help countries realize the transformational potential of digital identification systems to achieve the Sustainable Development Goals. It operates across the World Bank Group with global practices and units working on digital development, social protection, health, financial inclusion, governance, gender, legal, among others.

The mission of ID4D is for all people to be able to access services and exercise their rights, enabled by inclusive and trusted digital identification systems. ID4D makes this happen through its three pillars of work:

- Thought leadership and analytics to generate evidence and fill knowledge gaps;
- Global platforms and convening to amplify good practices, collaborate and raise awareness; and
- Country and regional engagement to provide financial and technical assistance for the implementation of robust, inclusive and responsible digital identification systems that are integrated with civil registration.

The work of ID4D is made possible through support from the World Bank Group, the Bill & Melinda Gates Foundation, the Australian Government, the UK Government, the French Government and the Omidyar Network.

To find out more about ID4D, visit id4d.worldbank.org. To participate in the conversation on social media, use the hashtag #ID4D.

Acknowledgments

This report was prepared by Tina George, Anita Mittal, Vasumathi Anandan and Ines Rodriguez Caillava, as part of the Identification for Development (ID4D) Initiative, the World Bank Group's cross-sectoral effort to support progress toward identification systems using 21st century solutions, and the Social Protection & Jobs Global Practice. It was made possible through the generous support of the from the Rapid Social Response Fund and the ID4D Multi-Donor Trust Fund (Bill & Melinda Gates Foundation, the UK Government, the Australian Government and the Omidyar Network).

This report benefitted greatly from the inputs by the World Bank Group staff including David Satola and Julia Clark under the supervision of Vyjayanti Desai.

The report would not have been possible without the insights and reviews by representatives from the Bill & Melinda Gates Foundation, Estonia's eGovernance Academy, India's Aadhaar program, Plan International, and the Omidyar Network.

Abbreviations

API	Application programming interface
DevOps	Development and Operations
ERP	Enterprise resource planning
EUPL	European Union Public License
FLOSS	Free/libre open source software
FOSS	Free and open source software
GFDRR	Global Facility for Disaster Reduction and Recovery
GPL	GNU General-Public License
ID	Identification
ID4D	Identification for Development
ITU	International Telecommunication Union
MPL	Mozilla Public License
ODbL	Open Database License
OSS	Open source software
PII	Personally identifiable information
PIN	Personal identification number
RFP	Request for proposals
SPJ	Social Protection and Jobs
ROI	Return on investment
SPB	Software Público Brasileiro

Introduction

This technical note is intended to contribute to understanding of how to leverage open source software (OSS) for global public goods[1] particularly in resource-constrained environments. The aim is to enable a more deliberate approach to building information systems that can serve as a global public good, rather than reinventing the wheel every time. Despite business processes being largely the same in various country contexts, each new project is typically built from scratch, as if there were no templates, code libraries or models, or lessons learned on which to base new implementations. Implementations in some domains are dominated by a few IT vendors that present significant switching costs and lock-in to governments that are already resource constrained. OSS solutions have the potential to address the challenges mentioned above and facilitate efficiency, robustness, security, and interoperability of information systems.

Governments in the digital age are interested to learn how OSS solutions can help build open, robust, interoperable, and secure service delivery platforms. Digital technology is increasingly the way citizens interact with government. From submitting passport applications to paying parking tickets and registering for social assistance, prior in-person interactions are now occurring online. For governments, modern identification (ID) systems allow for more efficient and transparent administration and service delivery, a reduction in fraud and leakage related to transfers and benefits payments, increased security, accurate vital statistics for planning purposes, and greater capacity to respond to disasters and epidemics. Equally important, social protection systems, programs, and policies help buffer individuals from shocks and equip them to improve their livelihoods and create opportunities to build a better life for themselves and their families.

While governments and public institutions recognize the value of digital technology, they still grapple with its implications for governance and service delivery. Governments often have good intentions and valuable products and services to offer citizens but engaging in digital transformation initiatives in low capacity and resource-constrained environments can be frustrating. Implementations of large and complex public sector IT systems have diverse stakeholders, and take time, financing, procurement, and rigorous testing. Even then, many projects do not work as planned, may have a large number of defects, performance issues, cost and time overrun, and are poorly received by users, besides being difficult and expensive to maintain. Governments find it difficult to build digital platforms for public service delivery, unlike private sector giants like Google, Amazon, or Facebook which successfully serve millions of consumers (Karippacheril and Tavoulareas 2014).

1 Digital public goods are tools that serve to educate us, help us thrive in our professional lives, enrich our cultural experiences, and ultimately do good for the benefit of humankind. Examples of these goods exist all around us in the areas of information, education, healthcare, finance, and more. Many also serve to further the UN's 2030 Sustainable Development Goals (**https://digitalpublicgoods.net/**).

Private sector service providers, Google, Amazon, and Facebook, serve high volume mass markets globally with 24x7 availability, making significant savings on technology, including software license fees. These firms use a mix of proprietary and OSS solutions to run complex infrastructure applications and database technologies, investing in in-house development and operations (DevOps)[2] skills, and driving down the total cost of ownership of their information systems.

Public sector service providers are studying open source technologies and standards with great interest, enticed by the promise of cost savings from software license fees and freedom from vendor lock-in. Government contracts and public expenditure have traditionally been dominated by big IT vendors (Public Sector Executive 2016). In 2010, a study on government open source policies by the Center for Strategic and International Studies (2010) revealed a total of three hundred and sixty-four open source policy initiatives across the world. Three examples of countries that have notably invested in OSS solutions for the public sector are the Republic of Korea, Brazil, and India. Korea built most of its public administration and digital governance systems using open source tools, partnering closely with local private sector IT vendors. Brazil has used OSS solutions for financial management and extensively for digital government solutions for the public sector. India has published open standards for digital government and its flagship digital identification program, Aadhaar, uses open source technologies and standards.

2 DevOps is a set of modern practices which seeks to closely bring together software developers and operations staff to work on the same project in a more collaborative manner. DevOps hopes to save time, money, and lead to shorter development cycles.

Objective and Scope

The objective[3] of this technical note is to serve as a source of technical reference on the question of leveraging an open source software model to develop a public good for public service delivery systems. The note is aimed at practitioners of digital government service delivery systems. First, a few key terms are defined, and then the benefits and the complexities of using an open source model to build public service delivery systems are discussed, reflecting on key considerations such as security, 24x7 support, interoperability, robustness, and so on. Finally, this note presents some options and a tiered strategy to support the development of an OSS solution as a global public good. In the appendix, a brief summary of experiences with reference to open source software from India, Brazil, and Korea is presented. It also includes brief summaries of the use of some open source software projects (GeoNode, MOSIP, and X Road) used in various country eGovernance projects.

Note that this paper does not elaborate an open source technology stack, such as choice of operating systems, web servers, application servers, database management, and so on. It is assumed that the technology used to develop a core OSS solution will be designed to be hosted or deployed over open source and proprietary platforms (operating systems, web servers, application servers and database servers) to provide choice and convenience to government agencies.

3 This paper was prepared in 2017 in the context of digital ID and social protection systems and has been subsequently revised to generalize it for all public service delivery systems. A case study on MOSIP, an open source platform for digital ID system that has been created in 2019 has also been included in the annexure.

Terminology

What is Open Source Software?

Open source software can be defined as software that is readily available with its source code and license, free of cost to anyone who wants to study, change, modify, or distribute it. Historically, it has usually been developed through collaboration or a group or informal network of programmers, who provide the entire source code to the end user.[4]

According to the Open Source Initiative's definition, OSS does not only imply access to source code, but also compliance with the following criteria for the terms of distribution (Open Source Initiative 2007):

- Free redistribution – license must not require royalty or any fee.
- Source code – program must allow free distribution of the source code.
- Derived works – license must allow modification and distribution.
- Integrity of the author's source code – license must explicitly permit distribution of software built from modified source code.
- No discrimination against persons or groups – license must not discriminate against any person or group of persons.
- No discrimination against field of endeavor – license must not restrict anyone from making use of the program in a specific field of endeavor
- Distribution of license – rights attached to the program must allow distribution.
- License must not be specific to a product – rights attached to the program must not depend on the program being part of a software distribution.
- License must not restrict other software – license must not place restrictions on other software that is distributed along with the licensed software.
- License must be technology neutral – no provision of the license may be predicated on any individual technology or style of interface.

4 Source code is the medium in which programmers create and modify software. It is essentially the text listing of commands to be compiled or assembled into an executable computer program.

Some commonly used terms for OSS include:

a. **Free and Open Source Software (FOSS)** – As the acronym suggests, FOSS refers to free and open source software that is provided to the user to copy, exchange, share, and use (UNESCO 2012).

b. **Free/Libre Open Source software (FLOSS)** – FLOSS is similar to FOSS but allows more freedom to edit/modify and distribute the software in original or modified version without any restrictions. FLOSS emphasizes the value of freedom, that is, with few or no restrictions, and it encourages the modification and redistribution of the source code.

What is Proprietary Software?

Proprietary software generally requires purchase of a license to use by payment of a one-time fee or recurring fees, and the source code is typically hidden from users. Proprietary software is also called closed-source software or commercial software. The copyright limits use, distribution, and modification, imposed by the copyright holder's publisher, vendor, or developer. Proprietary software remains the property of its owner/creator and is used by end users under predefined conditions usually defined in a license (Techopedia 2016). OSS source code is available free of cost to all, whereas proprietary software is not (Crooke 2016).

What Kind of Licenses do Open Source Solutions Use?

There is no one universally agreed-upon definition of FOSS software and various groups maintain approved lists of licenses. The Open Source Initiative (OSI) is one such organization keeping a directory of open-source licenses. The Free Software Foundation (FSF) maintains a list of what it considers free.[5] Some of the FOSS listed below illustrate the differences and nuances of different FOSS licenses.

- **GNU General-Public License (GPL) v3** is the most widely used free software license, which guarantees end users (individuals, organizations, or companies) the freedoms to run, study, share (copy), and modify the software. Permissions of this strong copyleft license are conditioned on making available the complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights.[6]

- **Apache License 2.0** is an open source software license released by the Apache Software Foundation (ASF). A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.[7]

5 https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses.
6 https://choosealicense.com/licenses/gpl-3.0/.
7 https://choosealicense.com/licenses/apache-2.0/.

- **Mozilla Public License (MPL) 2.0** is a simple copyleft license. Permissions of this weak copyleft license are conditioned on making available source code of licensed files and modifications of those files under the same license (or in certain cases, one of the GNU licenses). Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.[8]
- **European Union Public License (EUPL)** is a free software license created on the initiative of, and approved by, the European Commission (European Commission 2017). The EUPL is consistent with the copyright laws across the Member States of the European Union, and retains compatibility with popular OSS licenses such as the GPL.[9]

What are Open Standards?

There are several definitions of open standards. According to the International Telecommunication Union (ITU), open standards are those made available to the public, and are developed, approved, and maintained through a collaborative and consensus-driven process (International Telecommunication Union 2017). Complying with open standards is purely voluntary. However, some countries, such as India, have mandated the use of open standards for digital government services.

One of the principles of digital development is to encourage practitioners to adopt and expand existing open standards, invest in software as a public good, develop software to be open source by default, with the code made available in public repositories and supported through developer communities. Similarly, one of the ten main Principles of Identification for Sustainable Development emphasizes that governments should use open standards and ensure vendor neutrality (World Bank 2018).

Open standards are used by both OSS and proprietary software to enable interoperability with a variety of vendors. However, OSS built on proprietary standards cannot be distributed for free as the distribution would entail royalty payments for the standard in accordance with the licensing terms.

8 https://choosealicense.com/licenses/mpl-2.0/.
9 https://en.wikipedia.org/wiki/European_Union_Public_Licence.

Building an Open Source Software Solution

While some open source solutions cater to the public sector, commoditized open source software solutions to deliver critical applications, such as digital identity and social protection services, are few and far between.[10] One of the major objectives of the free and open source communities has been to standardize and commoditize software applications that are based on ubiquitous processes. These present a significant opportunity for the efficient utilization of resources (through cost savings from software licenses and freedom from vendor lock-in), fast tracking systems implementation, better integration, and interoperability of systems. For instance, Afghanistan adopted Odoo (NetLinks 2015) (previously called OpenERP) to streamline and automate their human resource management processes (Ghyasi 2015). Odoo is an OSS enterprise resource planning (ERP) software that commoditizes a range of ubiquitous business management functions such as human resource management, billing, accounting, manufacturing, and so on, and has technical support teams in 90 countries around the world. Odoo was adapted to support English, Pashtu, and Dar languages, with the assistance of a systems integration service provider based in India.

Benefits of an Open Source Software Solution

There are several reasons why open source software solutions are generally of interest to developing countries. The availability of source code of open source software allows it to be adapted to local needs forming the basis for viable local businesses, as the needs of users are often complex and contexts can vary dramatically from country to country. This feature further encourages enthusiasts from global communities to contribute towards innovation and facilitates technology transfer (Ghosh 2004).

Building OSS could facilitate the delivery of programs, save program costs, prevent vendor lock-in, enable integration and interoperability across government, engender trust in robust and secure systems, enable continuous innovation and enhance usability, localization, and citizen-centered design:

- **Expedite Program Delivery –** Governments spend a significant amount of time and effort conceptualizing, designing, developing, and testing software solutions to deliver public sector services. If a core OSS solution is available as a "public good," governments can fast-track the delivery of services using that core system. Value added services may be developed on top of the core system by the government or by third-party service providers since the specifications and source code are public. One example of this approach is the Android operating system. As an open

10 For example, the World Food Programme's SCOPE is a proprietary solution that enables both beneficiary operations management and digital ID. The software may be installed for free. However, the source code for the software is owned by WFP and cannot be modified by governments using the solution.

source core systems platform, it is used by several mobile device manufacturers to develop new features for devices and by third-party application developers to develop new mobile apps. End users benefit from a lower-cost device (bundled with a free and open source operating system) and can choose from a wide range of mobile apps available through the Google Play app store.

- **Save on Program Costs –** Designing software solutions entails significant effort on the part of governments to procure consulting and implementation services, and to work closely with the consultants to help define, review, and approve functional and technical requirements, in addition to implementing those software solutions. The procurement of vendors and development of systems in partnership with vendors often goes over budget and over time. One of the advantages of having a core OSS solution available as a public good is replicability (Gray and Satola n.d.). Governments can potentially save on costs by not having to "reinvent the wheel", including procuring consultants to help design and develop basic functionalities for the system or to automate standardized business processes without wasteful, duplicative funding. Rather, program costs could be focused on procuring and implementing value added services and system integration services to customize the core OSS solution to the specific country context. These could include enhancing security or other controls for deployment, training, operations, maintenance, and support. As an example, the Disaster, Risk and Recovery practice (GFDRR) of the World Bank Group estimated a return on investment (ROI) of 200 percent over the past seven years of deploying GeoNode, a free and open source software for online sharing of geospatial data (GFDRR 2017).

- **Realize Government Ownership over Software and Data –** As stated earlier, proprietary software solutions hide the source code from the buyer. This means that governments are typically beholden to the vendor in terms of control and ownership of the core software solution. Furthermore, a lack of attention to contractual clauses and fine print during the procurement process could lead to vendor lock-in. Ownership and control over mission-critical software program code remains with the vendor, which makes it difficult, if not impossible, for the government to switch to another vendor for support, even if the solution no longer meets their needs or if the cost of support has become unsustainable. Moreover, stored data may be subject to the whims of the vendor or subject to being lost if the vendor goes out of business (Gray and Satola n.d.). Transitioning support services to another vendor would be risky, and hence the term "vendor lock-in," on account of the steep learning curve for any other vendor that might be contracted to support the system in accordance with a service level agreement. The challenge of vendor lock-in has become a commonplace refrain in the public sector, particularly when proprietary software solutions are used. Inevitably, public sector agencies tend to customize these solutions to their specific needs, by appending new code on top of the source code. This is not to say that such a situation could not be envisaged if an OSS solution were used. Because an OSS solution makes the source code available, third-party service providers and other vendors compete to develop useful enhancements and to provide critical support services to the core solution. Moreover, skilled personnel within government are motivated to build internal technical capacity to maintain the OSS solution.

- **Enable integration and interoperability –** When government agencies design software solutions based on open standards, it becomes easier to enable interoperability and integration with solutions developed by other agencies that are also built in compliance with those standards, thereby enabling a "whole of government" approach to delivering services. For instance, designing a digital ID system, based on open standards at the national as well as the regional level, can help intergovernmental and regional cooperation in delivering crossborder travel and digital government services based on mutual recognition of each country's ID. Technology service providers such as biometric devices, smart card readers, and so on, based on open standards could work seamlessly (plug and play model) with ID systems at the national and regional level. Government service providers of social protection, healthcare, education, and financial services could work seamlessly with a digital ID system using the authentication services of a core OSS solution if it were built on open standards. Using open standards to build OSS solutions could increase market competitiveness and enable economies of scale for service providers. For example, Open Portal Guard (Municipality of Grosseto 2005) is an e-ID system built by the Ministry of Interior in Italy as an OSS and complying with open standards (Open Source Observatory and Repository 2009). While the project initially faced challenges in interoperability and standardization, their access control system[11] is now able to read e-ID cards from all over Italy as well as several other EU countries. An important factor that contributed to the success of the Open Portal Guard was that it had a team that was familiar with the open source ecosystem.

- **Engender Trust in Robust and Secure Systems –** Digital ID and social protection systems are among a group of key digital government systems that collect, store, and manage personal and confidential data on individuals, including nationals and nonnationals. These systems contain critical data for making decisions on the delivery of government services and often result in financial transactions, thereby placing rigorous demands on systems availability and performance, robustness, privacy, confidentiality, and security. One of the advantages of using a well-documented OSS is that countries can develop additional security measures and features on top of what is already provided by the core solution. By taking greater ownership for information security, considering inputs from various experts across government, the OSS could be benchmarked, tested, certified, and audited periodically by a third-party certification agency for security, performance, and availability. Moreover, as several organizations and countries progressively adopt the OSS, the product will benefit from enhancements and performance improvements made on the basis of knowledge, lessons, and experiences gained in various contexts. Such an approach could boost the confidence of governments in rolling out their systems, rather than delaying implementations for fear of failure and an irrevocable loss of reputation and trust. Community participation in strengthening OSS solutions can help address bugs, security holes, and fixes so that governments may deploy robust, secure, and high performance systems that are trusted by citizens.

11 "Access control system" refers to the entire spectrum of functionalities to perform authorization identification, authentication, access approval, and verifying the accountability of entities through login credentials such as PINs, passwords, electronic keys, and biometric authentication.

- **Continuous innovation –** As technology innovations, such as cloud computing, blockchain, and machine learning, continue to disrupt existing methods of operations and management, governments face a formidable challenge in adapting to those changes. Migrating from legacy systems to newer technologies becomes a Herculean task. One of the advantages of a well-supported OSS solution with a strong community of contributors and third-party service providers is that a migration path can be developed for the OSS core solution as a "global public good" ensuring backward compatibility to legacy implementations of that system. Such an approach could enable the rapid adoption of technology innovations by governments and communities around the world.

- **Usability –** When OSS solutions were at an early stage, development was focused on functionality and cost efficiency rather than on the user interface/experience. However, in recent years, OSS solutions have turned their attention to human-centered design without sacrificing cost efficiency and functionality. A design sensibility aimed towards simplicity and accessibility of the solution to the extremes (for example, older people, disabled people, among others) rather than the middle (for example, able people) enable technology acceptance, adoption, and diffusion to a greater swathe of the population. For instance, Maua (2013) observes that in the case of Kenya, use of open source software projects need to adapt to produce systems that can be used by typical and nontechnical users. Technical knowledge is needed to make the process work correctly and efficiently. Programmers who work on open source solutions should be encouraged to consider the issue of usability and where possible to coalesce their systems.

- **Localization –** As the source code is open, it can be customized to accommodate local language requirements. Such an option may not be commercially viable for vendors of proprietary software. Small, local firms are in a better position to compete for procurement contracts, breaking entrenched monopolies in the IT sector and reliance on expensive technical experts from abroad. OSS also diminishes barriers to entry to the global software marketplace for software engineers from developing countries (Gray and Satola n.d.).

Challenges to Building an Open Source Software Solution

Development agencies and governments across the globe are attracted by the value proposition of OSS solutions and have initiated projects using open source solutions, tools, components, or standards. For instance, Tanzania has embraced free and open source software (FOSS) and practitioners observe many benefits as a result. However, they continue to face challenges that characterize the OSS domain, such as lack of support from vendors, information security, indirect costs, and usability (Oreku and Mtenzi 2013).

While some initiatives have been successful, and others moderately successful, several initiatives have failed.[12] This is not to say that hopes of building an OSS solution for public service delivery systems must be abandoned, but to consider lessons learned from other initiatives. The harsh reality is that the success of

12 A familiar example is the case of Microsoft's Office productivity tools, which are available as Apache Open Office based on the Open Document Format, an ISO/IEC standard, free of cost. However, Microsoft Office, having already gained a critical user-base, continues to dominate the marketplace. Nevertheless, it may be hard to extrapolate the case of mass-market consumer software such as Microsoft Office to niche solutions such as digital identity or social protection service delivery, where the end-users are public sector staff and government agencies.

an OSS solution largely depends on addressing challenges such as the most appropriate software license, a thriving ecosystem for systems integration, technical support and services, stakeholder resistance to change, and maintaining market competitiveness. These are detailed below:

- **Fostering an Ecosystem of Partners for Technical Support and Services –** A critical undertaking is to grow a thriving ecosystem of market-driven, third-party service providers/partners that can provide systems integration and implementation services, 24x7 technical support for operations and maintenance, and intensive training, all the while supported by a trusted intermediary agency or an honest broker that can help grow and sustain the ecosystem for an OSS solution as a public good for digital identity or social protection:

- **An Honest Broker to Build a Core OSS Solution and Sustain the Ecosystem for Support Services –** An honest broker or a trusted intermediary agency would be an entity (for example, a vendor, organization, consortium, or foundation) that can oversee the design, development, and management of the core OSS solution. Examples of such an arrangement includes the Linux Foundation and Google (for Android).

- **Trusted Technical Community to Support the Core OSS Solution –** To ensure the sustainability of the core OSS solution, a key role for an honest broker would be to manage the DevOps through a trusted technical community (either as closed group or as an open group of contributors) who would support the core OSS solution at the global level. An honest broker would encourage and sustain the participation of a trusted technical community, to develop and distribute enhancements/patches in a timely manner to meet the requirements of clients, and to ensure compatibility of enhancements with deployed software code. As the source code for a system is available, and since the idea is to build software for public sector use, an honest broker would also act as a gatekeeper to ensure that the community that has access to the code or participates in improvements to the core OSS solution are known, trusted entities. This could help ensure that their work is audited and certified to mitigate potential issues of malicious tampering with the core software that will be used and trusted by government agencies around the world to provide digital identity and social protection services to their citizens.

- **24x7 Support for the Core OSS Solution –** One of the top challenges to implementing an OSS solution is that of 24x7 support for the software. It would be critical to invest in building local technical skills and capacity to support systems integration and implementation services for an OSS solution, in addition to encouraging the growth of third-party service providers around the world to provide technical support for implementation, maintenance, operations, and training programs:

    - ***Technical Skills for Systems Integration and Implementation –*** The OSS solution must have the support of local technical resources with the required skillset and knowledge to support systems integration and implementation services for the software, at competitive prices. Local technical resources who are well versed in the core OSS solution would not only help build value added services specific to those country contexts and needs, but they would also support maintenance and operations.

- ***Technical Support and Training –*** Proprietary software is typically bundled with a package of technical support and training. In the OSS space, there has been rapid expansion of firms that can provide such support for established solutions. However, such an ecosystem of support will need to be built up for a new OSS solution. Savings in software license fees to the government may be redirected towards training and technical support, to further stimulate and encourage the market.

- ***Third Party Support for Maintenance and Operations –*** To help configure, customize and deploy the core OSS solution, it would be beneficial to have third-party vendors around the world who can manage support and maintenance activities including enhancements, change requests, and bug fixes in accordance with service level agreements. The challenge would be to encourage the growth of a critical mass of vendors that can support the core OSS solution such that government agencies might to be able to procure services from the market at competitive prices.

- **Managing Stakeholder Resistance to Change –** There is still considerable skepticism among stakeholders with regard to the total cost of ownership, robustness, security, performance, and support services available for OSS solutions, in contrast to successful and competitive proprietary software. These solutions store and manage sensitive and confidential information on people (nationals as well as nonnationals), are required to be available at all times, and have critical national and information security, and financial implications. Government stakeholders accountable for the delivery of public sector services to the people based on these platforms are often more confident of tried and market-tested solutions rather than betting on a new and untested solution, even if it were offered free as a public good.

- **Maintaining Market Competitiveness –** It is essential to strike a balance between mandating open source solutions and encouraging innovation in the private sector through competitive processes. While developing a solution using the OSS model, an honest broker should take care to grow and sustain a competitive market for provision of private third-party solutions built on top of the platform. While, on the one hand, it may be perceived that an OSS solution for public service delivery systems may stifle the market for provision of such solutions by the private sector, on the other hand, the OSS solution might help kickstart the market for provision of value added services built on top of the OSS solution, as well as for technical support, training, and other opportunities. Furthermore, it could help encourage smaller firms to successfully bid for and execute government contracts, breaking the domination of a few big firms in the IT market. In Canada, interestingly, OSS cannot be mandated as the government is concerned about restricting the private sector market for the provision of those solutions.

When is it Appropriate to Build an OSS Solution?

- ***Standardized Business Processes and Workflows –*** When processes and workflows are fairly standard for a service, there is a good business case for a generic common solution because the value proposition of the OSS solution will depend on minimal customization. The social protection service delivery chain involves standardized processes, starting from outreach, intake and registration, assessment of needs and conditions, decision on enrollment, decision on benefits and service package, implementation of transactions, services or payments, and case management.

In the case of digital ID, the standard delivery process involves: registering individuals, issuing credentials, authenticating individuals, and managing digital identity data. A core OSS solution could be built to automate these standardized processes. The core OSS solution should be adaptable to country-specific configurations with minimal customization.

- ***Demand Driven –*** Stakeholders and counterparts from governments should appreciate the need and the value proposition in implementing an OSS solution. Prior to embarking on the development of a core OSS solution, a few pilot countries should be identified, who would be interested to implement the solution. Success in these pilot countries would create a demonstration effect and encourage others to follow suit. The more countries that implement the core OSS solution, the more robust the solution and the support ecosystem will become over time.

- ***Economics of Platforms –*** Platforms are defined "as building blocks (products, technologies or services) that act as a foundation upon which an array of firms (a business ecosystem) develop complementary products, technologies or services" (Gawer 2009). There are two requirements for a platform: first, it should perform a critical function of the overall system, or should solve a crucial technological issue, and second, it should be "easy to connect to," "build upon," and provide space for new and unplanned usage (Karippacheril, Nikayin, De Reuver, and Bouwman 2013). Two-sided or multisided platforms in economics refer to the mediating role of service platforms between two or more groups of agents (Evans, Hagiu, and Schmalense 2006; Rochet and Tirole 2003). The success of the core OSS solution as a "service platform"[13] will depend on ability to bring both sides of the platform on board at the same time – technology service providers such as smart cards and biometric devices, on the one hand, and government service providers, such as social protection services, financial services, health services, on the other hand – in order to create network externalities, and to motivate investments in terms of technical skills, financing, and so on, to deliver services. For example, the value proposition of a core OSS solution for digital ID should be compelling enough to smart card and biometric device manufacturers to adapt their technologies to plug and play. At the same time, other service providers should be compelled to utilize the functions of a core OSS solution for digital ID to authenticate citizens in order to deliver services.

What to Consider When Building an OSS Solution

- **Prototyping the OSS solution –** The requirements for any project are never complete and constantly evolve. Prototyping gives the participants an understanding of the product and helps in crystallizing the requirements. It also helps generate interest and confidence in the OSS solution leading to its adoption by the community. The total cost of software is insignificant in comparison to the cost of the project and, hence, investment in development of a prototype is helpful in several ways. A prototype may be developed rapidly, as shown by the experience of the Aadhaar team members in developing the Sunbird project. With the availability of cloud-based platforms, the upfront investment in development of prototype is minimal.

13 Platforms enable new services because of the reuse of platform components. They have lower fixed costs and enable shorter time to market for service providers. They can create opportunities for outside complementary providers. They are typically built upon a set of standards to ensure interoperability and compatibility between platform and complementary services. They typically offer application programming interfaces (APIs) or software development kits (SDKs) to enable third parties to develop services (Karippacheril, Nikayin, De Reuver, and Bouwman 2013).

- **Securing the OSS Solution –** The security of the OSS solution can be ensured by involving security experts from the design phase. Subsequently, hackers can be encouraged to break the system enabling curation of the product by fixing any loopholes identified. Hackathons are a great mechanism to cost effectively and innovatively utilize the crowdsourcing model to enhance the security and robustness of the OSS solution.

- **Microservice architecture –** A microservice architecture practice should be followed to ensure that multiple platforms are not created with different characteristics but rather code is reused by leveraging microservices for service orchestration. Each microservice would be like a small lego block (e.g. logging functionality) which is a tested block of code for functionality and performance and can be used by others for fast tracking the building of their subcomponents.

- **Quality Processes –** IT industry benchmarked quality processes should be followed during the development of the OSS solution to ensure the quality of the OSS, leading to an increased user base.

Additionally, a number of design principles (these are not exhaustive) may be considered during the development of an OSS solution:

- **Modular Architecture:** A modular architecture, where each module may be built by one or more providers with an open and interoperable interface as a standalone solution, provides choice and convenience to end users to build and configure their system by choosing modules from different providers based on their specific requirements.

- **Documented Open APIs:** A service-oriented architecture approach with open, published, documented APIs with relevant access controls can ensure the security of the system and data.

- **Scalability and Performance:** Scalable software that can meet the needs of smaller and larger countries in terms of population, will help improve performance and response time.

- **Security:** Data anonymization and cryptographic techniques, as well as other security features, such as robust internal access control, logging, auditability, traceability, and nonrepudiation for critical operations.

- **Privacy:** Many systems (for e.g. digital ID, financial, health and social protection systems) store and manage personally identifiable information (PII). The privacy of individual data must therefore be protected "by default" in these systems according to international standards such as fair information practices (FIPs). For example, data must be protected with access controls and there should be monitoring to prevent and/or detect a breach.

- **Flexibility, Configurability, and Extendibility:** The dominantly used options in each module should be built as configurable options in the software solution. Each country will be able to decide and configure the system to meet its requirements. In case the configuration is not a dominant requirement and is not a part of the software offering, the system should be well documented for the country to use the services of a system integrator to build its extensions and plug into the main system. The system should be flexible allowing mix and match with other components which the country may decide to use for legacy or other reasons.

A Tiered Approach to Building an OSS Solution as a Global Public Good

Prior to building a core OSS solution, there are other activities (or phases of work) that can be carried out in a tiered approach, in order to develop a public good for digital identity and social protection delivery systems. These are:

Phase 1 – Identification of technical standards and interoperability framework to develop interoperable, secure, and robust systems and to avoid vendor lock-in. Adopt technical standards (open standards to the extent feasible), guidelines, and frameworks for digital identity and social protection systems to help ensure that the systems developed by vendors, technology service providers, and government service providers are all interoperable.

Phase 2 – Business process mapping and definition of requirements (functional and technical) for these two distinct areas of operation, as a public good. Publish business process maps, functional requirements, and technical requirements online as an openly available and downloadable resource to help ensure that organizations do not reinvent the wheel, conducting requirements analysis and systems design. This document will be a living document incorporating lessons learnt from various implementations, and inputs from the field.

Phase 3 – A template request for proposals (RFP) for countries seeking to build digital identity and social protection service delivery solutions with specifications for robustness, quality, availability, security, interoperability, avoiding vendor lock-in, developed with inputs from experts, and available as a public good. The template can be adapted to specific country contexts. It could also include a checklist of lessons learned, challenges, good practices, and so on. With regard to procurement, the development partner should be neutral as to the choice of proprietary and open source technologies, allowing clients to procure an OSS solution if it is competitive and fits their needs (Gray and Satola n.d.).

Phase 4 – Develop a generic core solution as an OSS which can be adopted by countries with their country specific configurations and customizations, with the help of vendors and as a consortium partner with the appropriate licensing arrangements, taking into account all of the benefits and challenges involved in building such a platform.

Appendix. Case Studies

Brazil

Brazil's Public Software platform (Software Público Brasileiro) is an initiative of the Ministry of Planning, Development and Management. The initiative originated as a response to the lack of software sharing in the public sector. Starting in 1995, there were a few unsuccessful attempts to share software developed by the public sector. In 2005, a first step was made to make public software available by sharing the Cacic software (Automatic Configurator and Collector of Computational Information). The Brazilian Public Software platform was launched in 2007with the aim of creating an environment for sharing and collaboratively developing software.

The Brazilian Public Software (SPB) is a specific type of free software that meets the needs of modernization of the public administration at any level of government (federal, state, and municipal) and is shared without charge in the Brazilian Public Software Portal, resulting in the economy of public resources and constituting a beneficial resource for public administration and for society.

The Brazilian Public Software Portal already has 70 solutions aimed at various sectors. The available services are even accessed by other countries, such as Uruguay, Argentina, Portugal, Venezuela, Chile, and Paraguay. The portal has been consolidating itself as a software-sharing environment. This results in more streamlined management of IT resources and expenses, broadening of partnerships, and strengthening of free software policy in the public sector.

The main benefits of the Brazilian Public Software are summarized below.

- **Economy of resources:** In recent years, there has been a gradual increase in government spending on IT, especially those carried out by the Brazilian government in software acquisition and maintenance. The Federal Government's Integrated Financial Administration System (SIAFI) shows that the annual amount of federal public administration (APF) spending on software purchases between 2003 and 2009 increased by 227 percent, with an annual average of 32.43 percent. One of the main reasons related to APF's intentions to share systems through the SPB model is precisely the possibility of reducing costs, since it reduces efforts to develop new software, taking advantage of existing stable codes, saving time of production.

- **Independence of suppliers:** Through the use of SPB, there is no dependence on suppliers and consequent vendor lock-in. By adopting proprietary software, there is a great chance of generating dependence on specialized suppliers, who are the only ones able to modify the code of that contracted system. In this context, suppliers can charge higher prices since there is no competition. Through SPB, where there is a license that allows access and modification of the source code by anyone, this dependency is avoided. At any time, a bidding process allowing the hiring of companies other than those responsible for the original software development can be carried out. Thus, there is a stimulus to competition between suppliers, consequently improving the quality of services and reducing costs, benefiting society and government.

- **Safety:** It is difficult to know if software is secure when there is no access to the source code. The use of SPB eliminates this problem, since it adopts a general public license. Open source programs comply with the principle of transparency and allow full auditing. It allows the removal of doubtful sections, dangerous faults or even backdoors (a malicious way of leaving a hidden invasion path in the program, without arousing the user's distrust) and, as a direct consequence, brings more security.

- **Sharing knowledge:** Information and communication technologies are consolidating as a means of expressing knowledge, cultural expression, and economic transactions. In networked society, based on computer-based communication, it is not acceptable for the languages used in this communication to be controlled by only a few large companies. In the development of open source software such as the SPB, innovations are shared among all, allowing improvements to be adopted by anyone, so knowledge is always disseminated, helping small and medium-sized enterprises. Through the communities created around public software, there is a strong sharing of knowledge. The entire society has access to knowledge, whether it is a small municipality or a large federal government body.

India

The Government of India has undertaken many initiatives for promoting and fostering the adoption of free and open source software (FOSS) in view of various inherent advantages like increasing interoperability, developing local capacity/industry, reducing costs, conserving foreign exchange, achieving vendor independence, enabling localization, and reducing piracy/copyright infringements. A FOSS unit has been established under the R&D division in the Department of Electronics and Information Technology in the Ministry of Electronics and Information Technology to support research and development, human resource development, deployment, training, and support on FOSS tools and technologies.

One successful flagship project is BOSS (Bharat Operating System Solutions),[14] a GNU/Linux based localized operating system distribution that supports 18 Indian languages BOSS Support Centres that have been set up throughout India.[15] Millions of installations of BOSS and EduBOSS (an educational variant of BOSS) have been achieved so far.[16]

The Government of India has also set up the Open Technology Center (OTC) to spearhead the technology exploration and provision of support services for adoption of solutions, which are based on open source software and open standards, in various e-governance projects and applications under the National eGovernance Program (NeGP). OTC led the drafting of the government's policy for adoption of open source[17] and the Framework for Adoption of Open Source Software in e-Governance Systems[18] with recommendations on the software stack for applications. NeGP Mission Mode Projects for Road Transport, Public Distribution System(PDS), the e-Hospital system,[19] e-procurement system, Government Cloud Platform ("MeghDoot") and others have made extensive use of open source technologies.

14 http://www.bosslinux.in/.
15 http://bosslinux.in/support-centre.
16 http://meity.gov.in/content/foss-products.
17 http://egovstandards.gov.in/sites/default/files/Policy on Adoption of Open Source Software for Government of India.pdf.
18 http://egovstandards.gov.in/sites/default/files/Framework for Adoption of Open Source Software in e-Governance Systems.pdf.
19 https://ehospital.nic.in/ehospitalsso/.

Two examples of projects implemented by the Government of India which are similar to the model discussed in this paper for digital identity and social protection systems are outlined below:

- The Ministry of Road Transport[20] entrusted the National Informatics Centre (NIC, a government agency) to standardize and deploy software called VAHAN for vehicle registration and SARATHI for driving licenses and compilation of data with respect to vehicle registration and driving licenses of all the states in the State Register and National Register. VAHAN and SARATHI were developed as a common core solution to capture the functionalities as mandated by the Central Motor Vehicle Act, 1988 as well as state motor vehicle rules. There are 36 customized versions of the core product to suit the requirements of the states and territories of the Government of India.

- For the PDS project,[21] NIC developed a one-stop information portal for the Public Distribution System (PDS) for the Ministry of Consumer Affairs, Food and Public Distribution. India's states have variations in rules for the distribution of goods (that is, food grains, kerosene, and so on) for economically and socially disadvantaged people. NIC developed a common application module (CAS) which the states can use free of charge and deploy a customized version by using NIC's services or another vendor. The CAS software module can be hosted either on an open source software stack (JBOSS/PostGresSQL/Linux) or commercial systems (Oracle/Windows).

For both road transport and PDS the states are also free to develop their own systems without using the NIC developed modules. However, a set of standards and specifications for adherence by these systems have been set down to ensure interoperability of the state systems with the central system managed by NIC.

Republic of Korea

The Government of the Republic of Korea made a strategic choice to develop systems using an open source approach. The development of open source based frameworks reduced dependency on foreign companies, and on private technology vendors (which can limit flexibility and increase costs). The development of software based on open source software provided fast and effective implementation of results gained from business process reengineering (BPR) and information systems planning (ISP).

However, the development of such systems by various government agencies did lead to problems concerning interoperability during the integration process of different e-government systems. Initially, the open source approach did not have a standardized framework for government, which resulted in substantial costs regarding integration of software, processes, and systems across agencies. An emphasis on standardization is key to minimize costs associated with integration of software, processes, and systems across agencies.

Accordingly, the private sector helped the government develop a common standard called e-GovFrame to enable an open source software development ecosystem. The National Information Society Agency (NIA) and Ministry of Interior (MOI) also collaborated to develop e-GovFrame. By developing common standards, the government hoped to eschew proprietary systems to minimize the government's dependence on private software technology vendors who each have their own software framework.

20 https://parivahan.gov.in/sarathiservice/sarathiHomePublic.do.
21 https://pdsportal.nic.in/main.aspx.

The eGovFrame included a standardized set of software tools and a library of reusable OSS e-government components for application development, integration, maintenance, and reuse of applications. This e-GovFrame was launched in 2008. It was mandated for vendors applying to develop e-government applications through a request for proposal (RFP) process to use the e-GovFrame for software development. By 2014, more than 4,700 developers had been trained with more than 350,000 downloads. 450 e-government projects had used this framework with a budget of US$1.26 billion, significantly weakening vendor lock-in and adopting a flexible framework. Previously, 80 percent of government projects would be awarded to large vendors; more than 60 percent of the projects are now awarded to small vendors, enhancing the competitiveness of smaller vendors in implementing e-government.

GeoNode [GFDRR: 2017]

Starting in 2009, the Global Facility for Disaster Reduction and Recovery (GFDRR) of World Bank and its partners developed GeoNode. GeoNode is web-based, open source software that enables organizations to easily create catalogs of geospatial data, and that allows users to access, share, and visualize that data. Today, GeoNode is a public good relied on by hundreds of organizations around the world, and which receives continuously increasing investment from existing and new partners. These partners form the core of a thriving, mutually beneficial ecosystem of users and contributors – an ecosystem that includes nongovernmental organizations (NGOs), government agencies from a variety of countries, commercial participants, and motivated individuals.

GFDRR's direct and in-kind investment in GeoNode over the past six and a half years has been in the range of US$1-1.5 million. Partners have also made significant investments in GeoNode; a conservative estimate of these partner investments comes to about US$2 million over the same time period.

GFDRR's investment in GeoNode would be a reasonable amount even viewed strictly as a software development cost: the GeoNode software today represents return on investment of about 200 percent in terms of code written, since the current GeoNode project would most likely have cost US$2-3 million if GFDRR had produced it alone as proprietary software, without building an open source community around the codebase. The cost of licensing and configuring a commercial "off-the-shelf" proprietary solution would have been even greater, as the total cost would grow directly with the number of installations, while offering less long-term flexibility to meet the evolving needs of GFDRR and its partners.

However, the resultant software code is only part of the story. GFDRR and its partners have structured the project in ways that encourage participation by others who have similar needs, creating a self-sustaining open source community that functions independently of the continued presence of any particular long-term sponsor.

In particular, GFDRR followed these principles:

- **Simultaneously contract out and hire internally:** Outside developers increase the commercial viability and "social surface area" of the software project, while internal staff both contribute to developing the software and provide natural day-to-day oversight of the outside contractors.
- **Sponsor in-person events:** Partners met, learned, and collaborated at these events much more effectively than they could have if they had worked together only remotely.
- **Create partnerships:** GFDRR used staff time and connections to bring in peer institutions, which then invested in GeoNode themselves.

- **Train users:** GFDRR encouraged client countries to deploy GeoNode and invested in these deployments by allocating some staff time for training. Note: the numbers presented here do not include in-country training, only training at relevant events.

Beyond the considerable technical success of the GeoNode software, this open source ecosystem itself represents a significant return on investment for GFDRR and its partners.

With GeoNode now a self-sustaining project, GFDRR is able to reap the ongoing benefits of its continuing software development while shouldering very little of the costs. Instead, GFDRR is able to invest in building capacity of partner countries and user communities to deploy, maintain, and use GeoNode platforms. Furthermore, GFDRR – along with everyone else who uses GeoNode – receives the additional benefit of having a place in which to interact with other organizations who have similar needs. The GeoNode project is a living repository of best practices for geospatial data, a forum in which to find highly qualified geospatial specialists around the world, and a place to discover new collaborators in gathering, managing, sharing, and using geospatial data. The software itself is now easy enough to use so that people from local governments or universities can set up their own instances of GeoNode without assistance from or involvement by GFDRR.

This steady growth in number and diversity of participants is a hallmark of a flourishing open source project, and some elements of GFDRR's approach to GeoNode could be applied to other projects to achieve similar results. These best practices include, among others:

- Run as an open source project from the very beginning
- Engage other organizations commercially
- Focus on communications and evangelism early
- Find and encourage the right partners
- Invest in collaboration infrastructure
- Hold events and sponsor attendance
- Use funding choices as a signal to peer institutions
- Improve user experience to attract new users
- Change the nature of investments as needed

GeoNode's future as a public good seems secure. It is now used and maintained by hundreds of organizations – governmental, nonprofit, and commercial – and GFDRR can expect to benefit from the project for many years to come.

MOSIP

Introduction

Establishing an individual's identity is a complex task. Across the developing world, the arguments in favor of developing ID systems are broadly the same: without it, targeted welfare programs do not reach

their intended beneficiaries; the lack of an established identity prevents the most underprivileged from accessing a host of critical services, including access to healthcare and finance; and governments remain concerned about their ability to identify people uniquely for the purpose of holding free and fair elections (Gleb and Diofasi Metz 2018, 7-8).

Consequently, as of 2016, all but twelve of the world's low- and middle-income countries have launched national-level ID program, including every country in sub-Saharan Africa. While a few[22] have been successful, other countries have little to show despite substantial investments.

In many cases, this is in part due to several technology challenges that result from:

Vendor lock-in, and lack of interoperability: Most options available to national governments are proprietary, and use nonstandard protocols and components, making integration with other services and systems difficult (Desai, Gelb, Clark, and Diofasi 2017). Vendor lock-in often leaves countries reliant on proprietary technology which is costly and difficult to adapt over time.

High costs: Current options in implementing digital identity solutions can pose significant costs for governments, with on ID4Africa estimate suggesting costs up to US$240 million for a country with a population of 30 million people.[23] Factors contributing to the high costs include lack of technical capacity in government in designing and procuring IT systems, and bureaucratic procedures often unsuited to software development.

Foundational ID as a Modular Open Source Platform

Modular Open Source Identity Platform (MOSIP)[24] was conceived as a response to the challenges of this nature. Anchored at the International Institute of Information Technology, Bangalore (IIIT-B) as a global digital public good, and funded by the Bill & Melinda Gates Foundation, Omidyar Network, and Tata Trusts, MOSIP presents a different way of approaching the architecture, design, and integration of large-scale systems, one that recognizes ID systems as a strategic asset of a country.

MOSIP was created with the following objectives in mind:

- The development of a world-class open source digital ID Platform as a global digital public good, inspired by the Principles of Identification for Sustainable Development,[25] incorporating inclusiveness for universal accessibility; robust, secure, and sustainable design; and good governance to protect and empower people.
- Laying the foundation for the continuous evolution of MOSIP through a strong community of commercial partners and developers.
- Supporting the adoption of the MOSIP system in countries, to ensure the system benefits from design iterations and a constant feedback loop.
- Advance the global knowledge base and advocacy efforts towards open source solutions to the challenge of identification.

22 Kenya, Botswana, South Africa, and Zimbabwe (Gleb and Diofasi Metz 2018, note 2).
23 http://www.id4africa.com/2019/almanac/MOSIP.pdf
24 http://mosip.io.
25 Cite principles.

Platform Development

With the understanding that every country's needs and existing systems will vary, MOSIP follows a modular architecture, building the core modules needed for an identity system. These loosely coupled modules together enable the collection of demographic and (if applicable) biometric information from an individual, deduplication of the information using an external biometric deduplication engine (the ABIS or Automated Biometric Identification System), issuance of a unique ID, and authentication on the basis of that ID.

Key Design Principles

- **Privacy and Security:** MOSIP subscribes to the principle that the ID issuer is only the custodian of an individual data, and seeks to give individuals control over their data.
- **Open Source and Open Standards:** MOSIP has to coexist with a country's digital infrastructure. MOSIP fosters interoperability through its open source and open standards.
- **Modularity and Configurability:** MOSIP has independent and interchangeable modules with API-based implementation.
- **Scalability and Manageability:** MOSIP is horizontally scalable such that every component can individually scale up to meet varying load requirements. It is also designed with easy auditing, monitoring, testing, and upgrades in mind.

MOSIP Roadmap

MOSIP source code was released on GitHub (under the Mozilla Public License 2.0) in July 2019,[26] and it has a well-defined roadmap for the short, medium, and long term activities.

MOSIP Upkeep

For sustained momentum and active development on the platform, the MOSIP project plans to adopt contributions from countries adopting the platform, system integrators, research and university scholars as a regularly contributing base. Such contributions from diverse sources also need to be tested, certified, and incorporated into the product. Some of the options being considered to fund the small team to carry out these activities include using social capital, and revenues generated through support, consulting and ecosystem programs.

Country Adoption

The MOU between IIITB and Morocco was signed in August 2018 and with the Philippines in August, 2019 for their support in building their ID system based on MOSIP. Several other countries have expressed interest in the system.

26 https://github.com/mosip.

Ecosystem Development

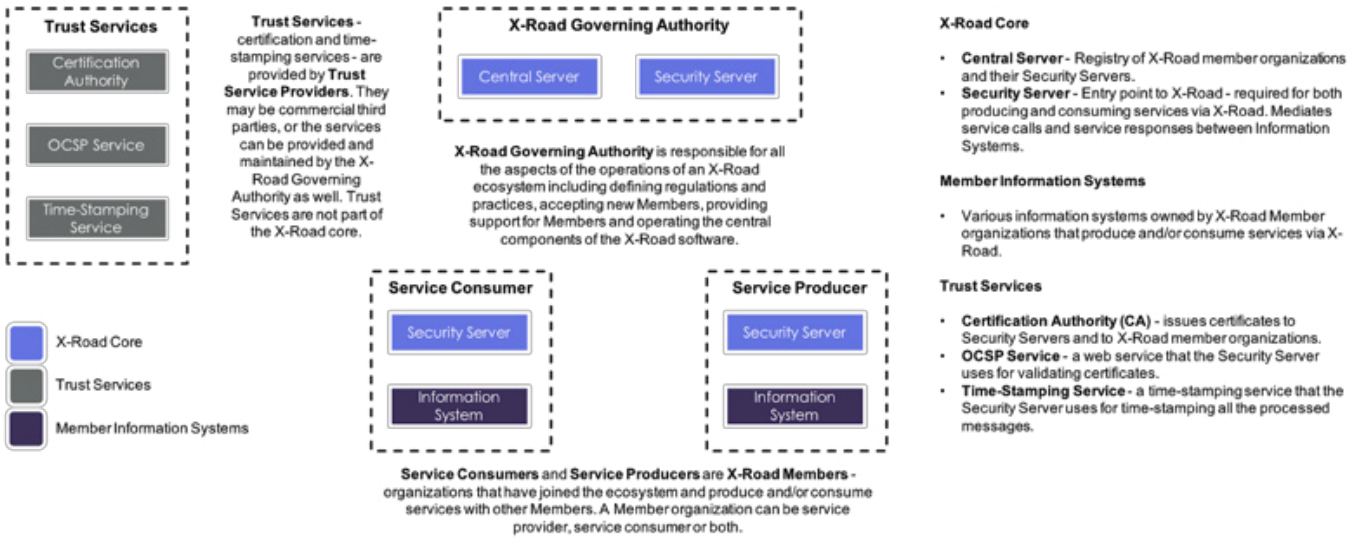
As MOSIP moves towards the adoption phase, it is important to nurture the community around the platform:

- **Technical Consultants:** Advising governments in digital transformation and guiding them through implementation.
- **Biometrics and Devices Companies:** In order to integrate with MOSIP, biometrics and devices companies have to build a connector in accordance with published specifications.
- **System Integrators:** Companies that package, market, and distribute MOSIP in a variety of contexts.
- **Open source community** that taps into existing developer groups, universities, and code contributions from the MOSIP commercial ecosystem.

X Road: A Secure Open Source Data Exchange Layer

X-Road[27] is an open source data exchange layer solution that enables organizations to exchange information over the Internet. It is a centrally managed, distributed data exchange layer between information systems and provides a standardized and secure way to produce and consume services. X-Road ensures confidentiality, integrity and interoperability between data exchange parties.

Figure A.1. X-Road Components, Roles and Responsibilities



Source: Petteri Kivimäki, 2019. https://www.apiscene.io/lifecycle/article-x-road-a-secure-open-source-data-exchange-layer. Used under Creative Commons CC BY 4.0 license. https://creativecommons.org/licenses/by/4.0/

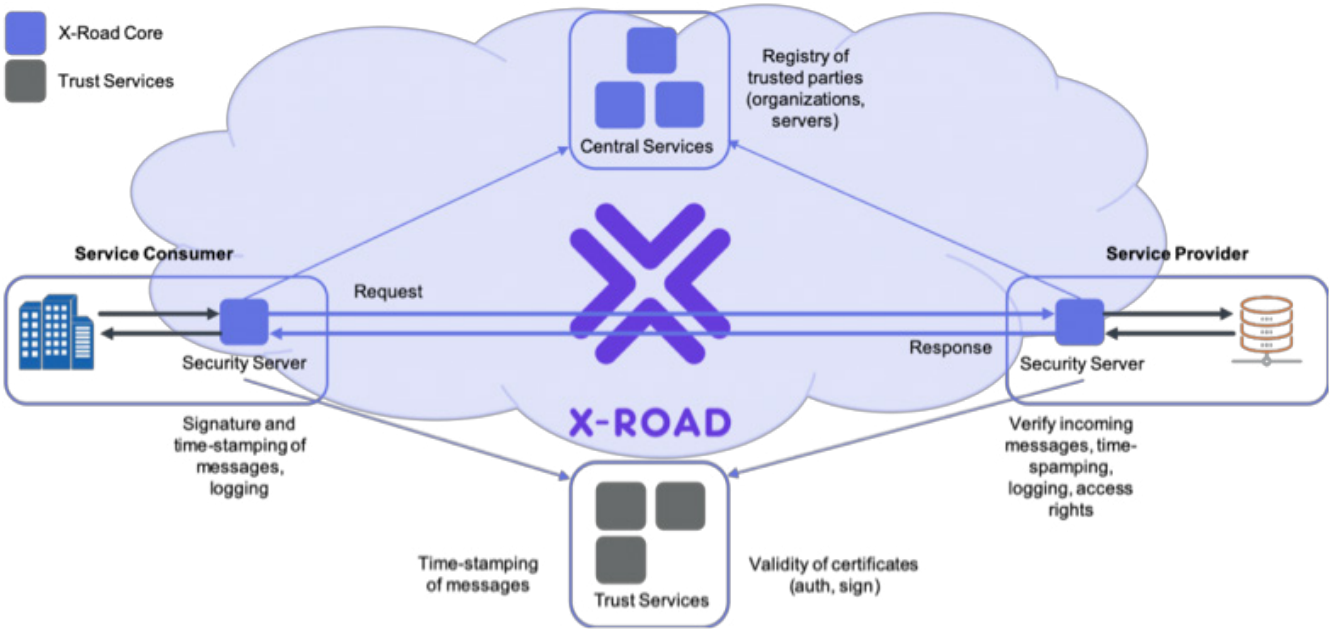
27 https://www.apiscene.io/lifecycle/article-x-road-a-secure-open-source-data-exchange-layer/.

X-Road is used nationwide in the Estonian public administration and in Finland's data Exchange Layer service (known as Soumi.fi). X-Road is released under the MIT license and is available free of charge for any individual or organization. The Nordic Institute for Interoperability Solutions (NIIS) is responsible for the development of the X-Road core and provides support and insights to the X-Road community.

Trusted Network

The identity of each organization and technical entry point (security server) is verified using certificates that are issued by a trusted certification authority (CA) when an organization joins an X-Road ecosystem. The identities are maintained centrally, but all the data is exchanged directly between a consumer and provider. Message routing is based on organization and service level identifiers that are mapped to physical network locations of the services by X-Road. All the evidence regarding the data exchange is stored locally by the data exchange parties, and no third parties have access to the data. Time-stamping and digital signature together guarantee nonrepudiation of the data sent via X-Road.

Figure A.2. X-Road Architecture

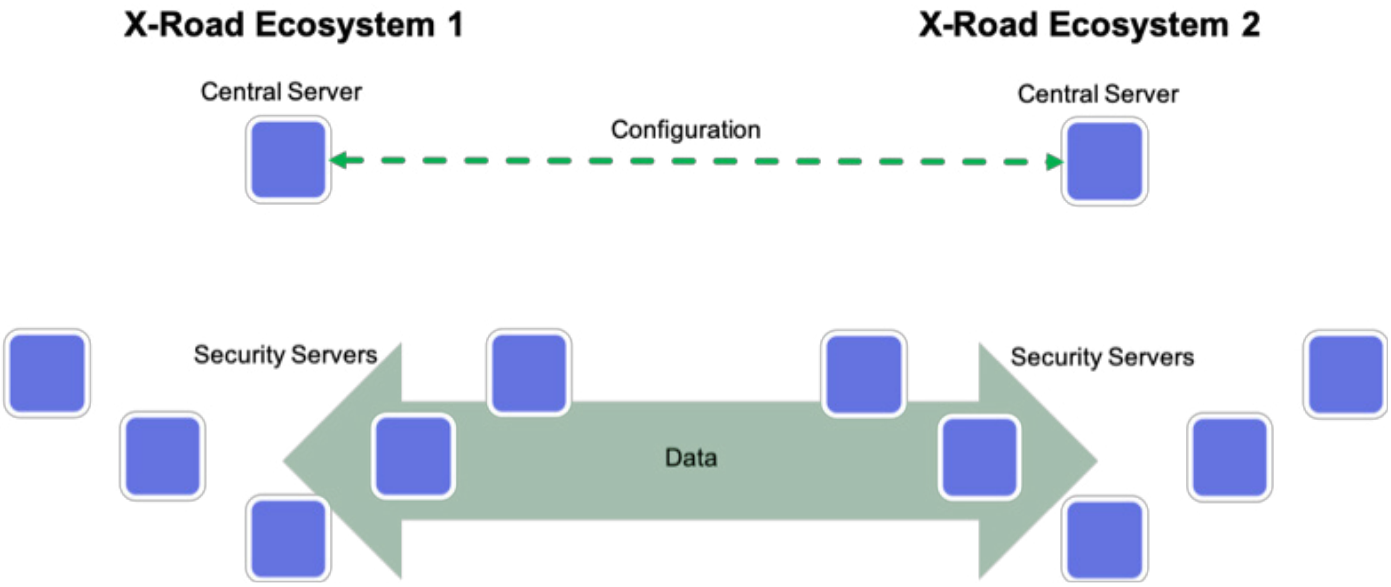


SSource: Petteri Kivimäki, 2019. https://www.apiscene.io/lifecycle/article-x-road-a-secure-open-source-data-exchange-layer. Used under Creative Commons CC BY 4.0 license. https://creativecommons.org/licenses/by/4.0/

An X-Road ecosystem is a community of organizations using the same instance of the X-Road software for producing and consuming services. The owner of the ecosystem, the governing authority, controls who is allowed to join the community, and the owner defines regulations and practices that the ecosystem must follow. The ecosystem may be nationwide, like in Estonia and Finland, or it may be limited to organizations matching certain criteria, for example, clients of a commercial service provider. Technically, the X-Road software does not set any limitations on the size of the ecosystem or to the member organizations.

Two X-Road ecosystems can be joined together, federated. Federation is a one-to-one relationship between two ecosystems. Members of federated ecosystems can publish and consume services with each other as if they were members of the same ecosystem. For example, Finland's and Estonia's data exchange layers are connected to one another which enables crossborder data exchange between the countries. Federation provides members of the federated ecosystems a single access point to both local and crossborder data sources. However, it must be noted that federation is not only about technology as both legal and administrative agreements are needed between the X-Road operators and member organizations that exchange data.

Figure A.3. X-Road Federation – Connection Between Two X-Road Ecosystems



Source: Petteri Kivimäki, 2019. https://www.apiscene.io/lifecycle/article-x-road-a-secure-open-source-data-exchange-layer. Used under Creative Commons CC BY 4.0 license. https://creativecommons.org/licenses/by/4.0/

X-Road is an open source technology, and anyone has access to it free of charge. An X-Road ecosystem is managed by a governing authority that controls who is allowed to join the ecosystem. In Estonia and Finland, the ecosystems are open for all kind of organizations (public, private, non-profit etc.) and joining them is free.

X-Road Development

NIIS is responsible for developing X-Road core technology, and welcomes everyone to submit source code contributions, new ideas and enhancement requests regarding X-Road. The backlog is public – anyone can access it, leave comments and submit enhancement requests through the X-Road Service Desk portal. Accessing the backlog and service desk requires creating an account which can be done in few seconds using the signup form, which is available on GitHub.[28] Today, it is implemented in Finland, Kyrgyzstan, the Faroe Islands, Iceland, Japan, and other countries. Similar technology that is based on the Estonian interoperability experience has also been implemented in Ukraine and Namibia.[29]

28 https://github.com/nordic-institute/X-Road.
29 https://e-estonia.com/solutions/interoperability-services/x-road/.

References

Center for Strategic and International Studies. 2010. *Government Open Source Policies*. Introductory Note. Washington, DC: Center for Strategic and International Studies. http://archive.openforumeurope.org/source/open-source-library/100416_Open_Source_Policies.pdf.

Crooke, D. 2016. "Open Source vs. Proprietary Software." Blog. *GreenNet*. https://www.greennet.org.uk/support/open-source-vs-proprietary-software.

Desai, Vyjayanti T., Alan, Gelb, Julia Clark, and Anna Diofasi. 2017. "Ten Principles on Identification for Sustainable Development." *Digital Development* (blog), February 7. http://blogs.worldbank.org/ic4d/ten-principles-identification-sustainable-development.

European Commission. 2017. Commission Implementing Decision (EU) 2017/ 863 of 18 May 2017 Updating the open source software license EUPL to further facilitate the sharing and reuse of software development by public administration. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2017.128.01.0059.01.ENG&toc=OJ:L:2017:128:TOC.

Evans, D. A., A. Hagiu, and R. Schmalense. 2006. *Invisible Engines How Software Platforms Drive Innovation and Transform Industries*. Cambridge, MA: MIT Press.

Gawer, A. 2009. "Platform Dynamics and Strategies: From Products to Services." In *Platforms, Markets and Innovation*, edited by A. Gawer. Cheltenham, UK: Edward Elgar.

GFDRR. 2017. *Open Data for Resilience Initiative & GeoNode: A Case Study on Institutional Investments in Open Source*. Washington, DC: GFDRR. https://opendri.org/resource/opendri-geonode-a-case-study-for-institutional-investments-in-open-source/.

Ghosh, R. 2004. The *Opportunities of Free/Libre/Open Source Software for Developing Countries*. Background Paper. University of Maastricht. https://www.iprsonline.org/unctadictsd/bellagio/docs/Gosh_Bellagio4.pdf.

Gleb, Alan and Anna Diofasi Metz. 2018. *Identification Revolution: Can Digital ID be Harnessed for Development? Washington, DC: Centre for Global Development*.

Ghyasi, F. 2015. "Enterprise Resource Planning (ERP) Using Odoo/Open ERP." Presentation. NetLinks. https://www.slideshare.net/fghyasi/enterprise-resource-planning-using-odooopenerp-in-afghanistan.

Gray, P. and D. Satola, D., n.d., Open Source Software - Alternatives to the Challenges of Software Procurement and Development. Discussion Paper. Washington, DC: World Bank Group.

International Telecommunication Union. 2017. *Definition of Open Standards*. Telecommunication Standardization Sector (ITU-T). Geneva: International Telecommunication Union. http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx.

Karippacheril, T.G. and E. Tavoulareas. 2014. "Getting Digital Service Delivery Right." *Governance for Development* (blog)., January 28. http://blogs.worldbank.org/governance/getting-digital-right.

Karippacheril, T.G., Nikayin, F., De Reuver, M., Bouwman, H. 2013. "Serving the Poor: Multisided Mobile Service Platforms, Openness, Competition, Collaboration and the Struggle for Leadership." *Telecommunications Policy* 37 (1): 24-34.

Maua, M. 2013. "Motivation for the Adoption of Open Source Software in Kenya Academic Libraries." Presentation. Pwani University. Kenya. https://www.researchgate.net/publication/261050507_Motivation_for_the_adoption_of_open_source_software_in_Kenya_academic_libraries

Municipality of Grosseto. 2005. "Open Portal Guard." Government website. Grosseto. http://openportalguard.sourceforge.net/.

NetLinks. 2015. "Why Odoo." Blog. NetLinks. https://www.netlinks.af/?p=7843.

Open Source Initiative. 2007. *Open Source Definitions*. Open Source Initiative. https://opensource.org/docs/osd.

Open Source Observatory and Repository. 2009. "Grosseto (Italy) develops the OpenPortalGuard e-ID system." Blog. European Commission. https://joinup.ec.europa.eu/community/epractice/case/grosseto-italy-develops-openportalguard-eid-system.

Oreku, G. and F. Mtenzi. 2013. "Adoption and diffusion of open source software in Tanzania: A way forward". 2013 IST-Africa Conference and Exhibition, IST-Africa 2013. 1-10. https://www.researchgate.net/publication/261054775_Adoption_and_diffusion_of_open_source_software_in_Tanzania_A_way_forward.

Public Sector Executive. 2016. "Open Source Software Finds It's Sweet Spot in the Public Sector." *Public Sector Executive*, June 15. http://www.publicsectorexecutive.com/Public-Sector-News/open-source-software-finds-its-sweet-spot-in-the-public-sector-.

Rochet, J.-C., and J. Tirole. 2003. "Platform Competition in Two-Sided Markets." *Journal of the European Economic Association* 1 (4): 990-1029.

Techopedia. 2016. "Proprietary software." Definitions. *Techopedia*. https://www.techopedia.com/definition/4333/proprietary-software.

UNESCO. 2012. "*Free and Open Source Software* (FOSS)." Conference Proceedings. Intergovernmental Council for the Information Program. http://en.unesco.org/foss.

World Bank. 2018. *Principles of Identification for Sustainable Development: Toward the Digital Age*. Washington, DC: World Bank Group. http://documents.worldbank.org/curated/en/213581486378184357/pdf/112614-REVISED-English-ID4D-IdentificationPrinciples.pdf.

id4d.worldbank.org