Provision of Consultancy Support for GIZ – OpenIMIS

OpenIMIS System architecture & technical roadmap

Draft for Discussion

May 2018
Draft Version 5

Patrick Ernst
MIS Consultant

giz Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH

Report Cover Page

Project title	OpenIMIS Technical Roadmap
Client	GIZ
Contractor	Patrick Ernst
Date of report	January, 2018

Contact

Position	Name	Email & Cell phone
Consultant	Patrick Ernst	mail@patrickernst.de +49 176 61602329
Project Coordinator	Alicia Spengler	alicia.spengler@giz.de

Version History

Document Status	Delivered	Feedback
Draft version 1	24/04/2017	Feedback from Alicia Spengler, Julius Murke
Draft version 2	09/05/2017	Additional feedback from Uwe Wahser (GIZ/ NHIF Kenya) Kelvin Hui (GIZ Cambodia), Carl Leitner (PATH), Knut Staring (UiO). Most of the feedback referred to the three system options. Several comments (Wahser, Hui, Leitner) hinted at the complexity of option 3 (DHIS2 based system architecture). Picking up these comments, two major stream of thoughts have been included: • Phased implementation approach, module development, proof of concepts • Approach for step-by-step transitioning from MS IMIS Implementation to OpenIMIS (Based on option 2 or 3) Feedback from Knut Staring on mapping of MS IMIS data objects to DHIS2 configuration objects.
Draft version 3	21/06/2017	An excerpt of draft version 3 - the section “technical analysis of MS IMIS” - was pre-released to GIZ Nepal and SwissTPH earlier in June, the feedback is included in this version 3. On this basis some technical inaccuracies were corrected. Some of the comments, which were of a more generic nature, describing and appreciating technical and functional approaches of IMIS were also included directly in the chapter. Other comments were already implicit part of some of the discussions in other chapters of the document. The full SwissTPH comments are available upon request.
Draft version 4	2018-01	Included July 2017 comments from Swiss TPH. Development of synergistic approach, combining advantages of 3 proposed options.
Draft version 5	2018-05	Included OpenIMIS Technical Workshop Comments from February 2018
Draft version 6	2018-06	Restructured and updated the document. While version 5 was more of a documentation of the discussion of options that lead to the technical roadmap, version 6 aims to focus on the results of the discussion, including the contributions of the technical workshop from 2018-02.

Acknowledgements

In addition to above mentioned individual reviewers, other individuals and organisations have contributed during personal meetings or phone conferences: DHIS2 team at University of Oslo (Jørn Bra, Kristin Bra, Ola Titlestad, Sundeep Sahay, Lars Øverland) BlueSqare (Martin van Aken, Nicolas de Boerman) and Thoughtworks (Karl Brown).

Table of contents

List of figures

List of tables

List of abbreviations

API	Application Program Interface
CHF	Community Health Fund (Tanzania)
CRM	Customer Relationship Management
DHIS2	District Health Information System Version 2
EHR	Electronic Health Record
EMR	Electronic Medical Record
FHIR	Fast Health Interoperability Resources
GIS	Geographic Information System
GIZ	Gesellschaft für Internationale Zusammenarbeit
HL7	Health Level Seven
HMIS	Health Management Information System
ID	Identification
IT	Information Technology
LMIS	Logistics Management Information System
NHIF	National Health Insurance Fund (Tanzania) National Hospital Insurance Fund (Kenya)
IMIS	Insurance Management Information System
MoH	Ministry of Health
MoU	Memorandum of Understanding
OpenHIMIS	Open Health Insurance Management Information System (transitory naming for openIMIS in 2017)
OpenIMIS	Open Insurance Management Information System
OpenLMIS	Open Logistics Management Information System
OpenMRS	Open Medical Record System
openRBF	Open Result Based Financing
RfC	Request for Change
SDC	Swiss Agency for Development
Swiss TPH	Swiss Tropical and Public Health Institute
TRM	System Architecture and Technical Roadmap
UiO	University of Oslo
WHO	World Health Organization
XML	eXtensible Mark-up Language

1 Executive summary

This "System Architecture and Technical Roadmap" document serves to describe the future design of OpenIMIS. It is based on options that have been translated into a rough development and implementation roadmap. This also includes the possibility to transition step-by-step the countries currently using IMIS to the new version.

The application should be built based on certain **principles**, which were agreed upon during the inception workshop in 2016, and have been further detailed since. They include using open source technology to avoid license costs, to foresee scalability to support also larger countries and to plan for interoperability, creating a system that is modularized and can communicate with other systems through clearly defined APIs. Furthermore adhering to data standards should prepare also for future interoperability scenarios, respecting data security aspects. In addition, components of existing standard software/platform solutions should be re-used to benefit from technical, functional and community synergies. Finally the new solution should be able to cater also to existing users and implementations by offering a system transition path.

An important aspect is the status of the **user requirements**. Before making significant software investments, the user requirements need to be well defined to make sure that the software functionality covers the current and future customer needs. We consider the requirements to be sufficiently well documented, stable and predictable to proceed. The current IMIS version contains functionality that has proven to be fitting to the needs of three target countries, largely corresponding to requirements defined by a group of countries in the JLN Collaborative Requirements Development Methodology (CRDM) in 2011/2012.

A key activity for developing the roadmap is a technical and functional analysis of the current Microsoft based IMIS (**MS IMIS**). The analysis starts off with an asset inventory, describing the completeness and quality of the technical and functional documentation. In general, documentation is quite thorough and complete, giving partners a technical understanding to adapt the software to their needs. What is missing are hands-on installation and implementation guidelines, an important basis for creating an ecosystem where several partners can assume responsibility in the implementation process.

The technical analysis shows how MS IMIS was originally developed for Tanzania, and then modified to accommodate further functionality from Cameroun and Nepal. Due to its functional flexibility, IMIS is coping well with the country requirements, but it also has some limitations regarding modularity and interoperability. This makes it difficult to expand IMIS in a modular way, restraining the possibilities of external partners to contribute new functions. Finally, the technology is tied to the Microsoft ecosystem, requiring a Microsoft operating system, databases and license costs.

As a response to these limitations, three alternative system architecture options were developed and discussed:

- **Option 1** seeks to stick to the current **MS IMIS,** exploring options how to upgrade it to overcome above discussed limitations.
- **Option 2** discusses **re-writing** entirely the application using open source technology, at the same time introducing a more modular design and using micro-services and possibly also existing library components, for example from HL7 FHIR.
- **Option 3** aims to build a system architecture using existing open source solutions, such as **DHIS2** or other components from the openHIE ecosystem.

Above described options were then discussed in the context of their resource needs. One of the general observation is that in any case, a minimum **permanent team of 3-5 staff** will be required to maintain the software and to support countries, independent of the chosen technical approach. When comparing the three options, some differences become apparent. Continuing with MS IMIS as the only system would have the lowest cost in the beginning, but will possibly lead to rising costs if the limits of the system architecture show. As opposed to that, working in the openHIE context promises important synergies due to the existing communities, but it will require more important initial investment in order to integrate new functionalities into the existing framework.

Taking into account these options and corresponding resource needs, a **phased development approach** is suggested.

In a next step or in parallel, **technical concepts** will be elaborated to verify specific assumptions of the three options:

- For MS IMIS, the concept should outline the technical upgrade path, especially how to create more modularity and interoperability
- For the open source re-write, the concept should clarify how innovative system elements (micro-services, pre-built standard libraries) can be integrated into a development and implementation timeline, overcoming the stability and quality problems that usually come with re-writes
- For the openHIE based system architecture, cooperation approaches with openHIE partners should be explored to maximise technical and budget synergies.

A focus is also to benefit from leveraging synergies between the three options. During the discussions, considerable advantages of all three options were identified. An implementation approach that seeks to **combine advantages of the different options** is to build on a gradual transition from MS IMIS to option 2 and 3, where MS IMIS assumes for a defined period or a specific setting a specified function in a new system architecture. As a consequence, this TRM also describes some of the migration challenges, that come with a modular development path, ensuring that both existing and future user organisations have always access to viable system options.

2 Introduction

The open source health insurance management information system (OpenIMIS) initiative seeks to provide a comprehensive health information system linking patient, provider and payer data. This system shall be managed by a dedicated software team and continuously improved by an open source software community. The management information system shall have a modular, interoperable build, allowing it to be adapted to the needs of a variety of low- and middle-income countries on their path to universal health coverage. The initiative also seeks to provide capacity development services and foster a community of practice for implementers and users

The open source approach has been chosen over a proprietary solution due to its many advantages including the efficiency gains of being able to build on other countries' experiences; the progress towards international standardization of health systems data; the avoidance of vendor lock-in; lower administrative and operational cost; (technical) support by an international community of practice; and lastly, the opportunity of capacity development in the areas of health system management and governance, evidence-based decision-making, and ICT systems for health (including software development).

The technological basis of the initiative is the Insurance Management Information System (IMIS), designed by the Swiss Tropical and Public Health Institute (Swiss TPH) and funded by the Swiss Agency for Development and Cooperation (SDC), who is the software license holder. IMIS was first implemented in 2012 as the IT backbone for district-based pre-payment schemes in Tanzania. The system was then adopted for a mutual health insurance scheme in Cameroon and since 2015 – supported by the German Development Cooperation – for operating Nepal's national health insurance scheme. IMIS has grown organically and has been adapted to different types of health financing systems moving towards UHC.

This "System Architecture and Technical Roadmap" (TRM) document serves to identify options for the future development of OpenIMIS. These options will then be translated into a rough development and implementation timeline. This also includes the possibility to transition the countries currently using IMIS to the new version.

3 Design principles

The future OpenIMIS technology and system architecture should follow certain principles:

1. It should be built using **open source** technology, avoiding license costs
2. It should be **scalable** to support also larger countries
3. **Interoperability and modularity**: It should have a modular structure and be able to communicate with other systems through clearly defined APIs
4. **Data standards**: Adhering to data standards prepares also for future interoperability scenarios, including data security
5. It should re-use components of **existing standard software**/platform solutions, to benefit from technical, functional and community synergies
6. It should take into account the ongoing support needs of existing **users** and implementations (Tanzania, Nepal, Cameroun)

In the next sections, the different principles are discussed in more detail.

3.1 Open source

In the space of global health software, open source approaches have gained considerable traction. One of their main advantages is that they can be developed and maintained without license fees. This refers to server operating systems, user clients, databases and the application itself. But although open source can be free, it still has a cost. Just as with commercial software, it is important to have a concept that foresees resources for the maintenance of all involved pieces of software at the different levels.

3.2 Scalability

Health insurance software applications can be amongst the most resource demanding applications in the global health space. The system ought to be able to respond to different implementation scenarios. On the one hand it should be able to quickly support simple usage scenarios, where a basic implementation is required, that can later grow to more elaborate usage. On the other hand, OpenIMIS should also be prepared to support large scale scenarios in the long run, for example a beneficiary group of more than 100 million insurees which may result in more than a billion yearly claimed line items.

3.3 Interoperability and modularity

National health system architectures in low resource settings are often marked by the existence of many independent systems. Vertical disease programs, such as vaccination, communicable disease surveillance, TB, HIV/Aids, Malaria often have different funding sources leading organisations to set up isolated systems and creating information silos. A similar tendency can be observed for functional approaches, such as logistics, human resources (health worker registries) finance management and social health insurance systems.

Since several years there has been considerable effort to overcome these approaches through interoperability. Interoperability builds on existing applications, tying them together through data exchange (which normally requires an active technical maintenance from both sides). In the area of health insurance a number of interfaces are relevant, offering potential for the streamlining of operations, improvement of data quality and gaining analytical insight:

- Receive updated reference or master data from central repository (Facility IDs, medical services and items, population ID data, …)
- Receive enrolment data and contribution payment information from mobile app
- Receive claiming data from claiming apps or medical record systems
- Share approved cumulated claim data to finance management systems for payment
- Share reporting data to National Health Management Information System (HMIS) for integrated reporting

A major approach for interoperability has been defined in the openHIE framework.

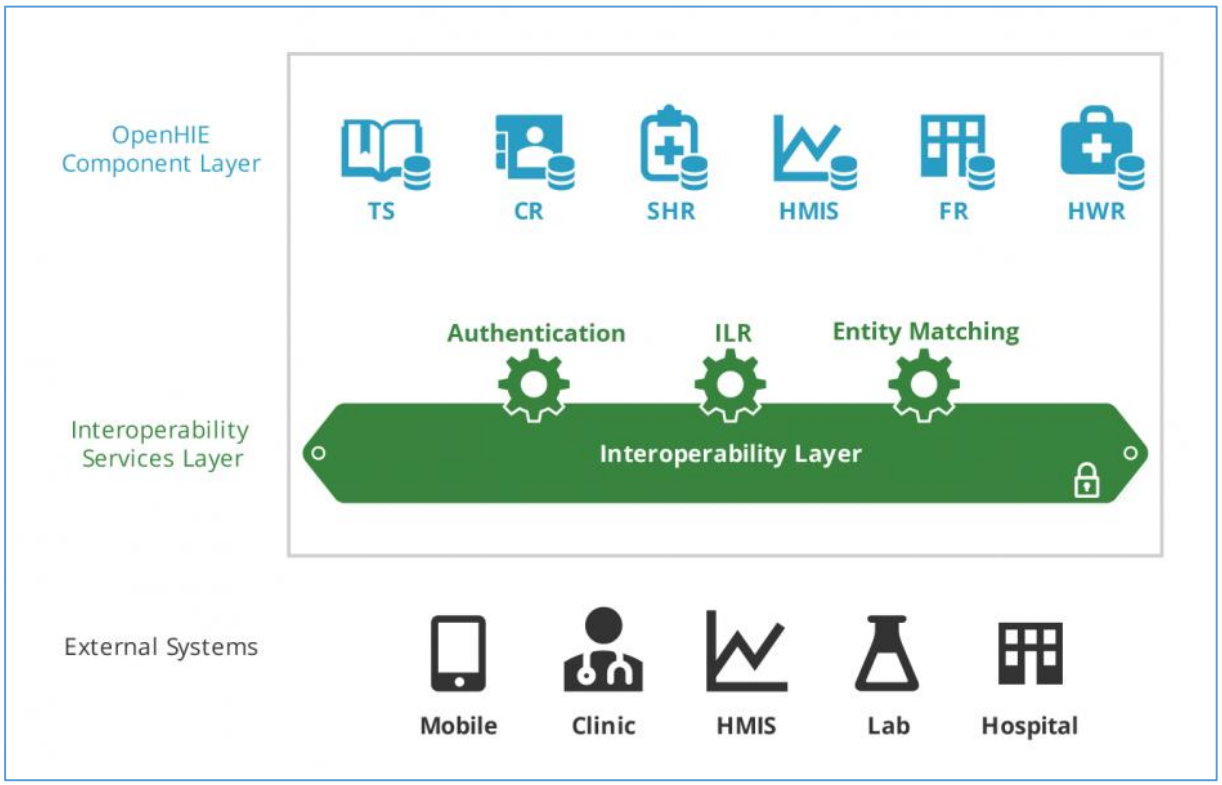

Figure 1: OpenHIE

This approach, which has recently been becoming the main reference for modelling national health system architectures aims to connect the different existing health software solutions into a connected architecture, proposing to standardise interfaces and exchange data through a dedicated system interoperability layers, that allows to exchange data connecting through a dedicated mapping layer.[1]

An important basis to build interoperable approaches is also to have a **modular** software design. This means that it is possible to modify components of one system without affecting the other component. This would allow another possibility to allow partners to develop additional functionality.

3.4 Adherence to standards

Whenever possible, the application should adhere to internationally established data standards. This is also an important basis to be prepared for future interoperability scenarios. An important data interoperability standard is **HL7 FHIR**. Supported by the large EMR and medical device makers, it is becoming the quasi standard for storage and interchange of data objects, such as a claim or basic patient data.

[1] It is interesting that in this OpenHIE design, Health Insurance is not explicitly mentioned. It may be useful to participate in the openHIE and advocate the explicit inclusion of Health Insurance System in the model, e.g. as an external system.

The application should also adhere to established standards of **data security**. The general challenges are that the application has to deal with two sensitive elements: Financial data and personal health data. In an interoperability context this is especially important, because when data is shared across systems and institutions, clear access rules need to be defined and enforced.

3.5 Build on existing software

By using components of **existing standard software/platform solutions**, costs can be reduced during development and especially during implementation and maintenance. Furthermore, the usage of existing components can help to connect to existing implementer and practitioner communities, which can accelerate adoption and integration with existing country solutions.

Over the last years, the health domain has seen an impressive growth of standard software solutions or flexible platforms, most of them open source, targeted at low resource settings. These systems have made implementations much more predictable, allowing countries to draw on the experiences of communities of practitioners and the availability of experienced implementers.

Some of the most important solutions in the health space are discussed below, with a special focus on their relevance for the health insurance sector.

Solution	Description	Relevance for OpenIMIS
DHIS2	Leading Health Information Management System (HMIS). While it started out managing mainly aggregate data, development focus has recently shifted to DHIS2 Tracker, which allows to capture event or patient related data. DHIS2 is used by Ministries of Health, NGOs and international agencies/donors. It is in use in an estimated >200.000 facilities in OpenIMIS target countries.	In this document, we will explore the option of building OpenIMIS in DHIS2 in greater detail. Although there are some challenges in the database structure of DHIS2, the wide usage of DHIS2 and its flexible platform approach make it an interesting option. A crucial asset of DHIS2 is its strong community, which counts hundreds of implementers and support staff in all target countries.
openMRS	Leading open source EMR in target countries. Used in almost 2.000 facilities. Tends to be used only in larger facilities (Specialized / National / Tertiary/University Hospitals), sometimes also in Regional or District Hospitals.	Larger Hospitals that are using openMRS may want to convert their encounter data into claims and transmit them electronically to OpenIMIS.
Bahmni	OpenMRS distribution that combines OpenMRS and Odoo (OpenERP) for Finance Management.	OpenIMIS can learn from closely integrating a Finance Management into the system set-up. Also the experiences from tightly bundling different solutions into an application package may be an interesting reference.
Resource Map	Specialized Health Facility Registry tool. Designed to feed facility master data to other systems. Used in ca. 5 countries.	OpenIMIS needs to receive updated facility metadata. In many countries, DHIS2 is the most reliable source of the facility data.
CommCare	Mobile data collection tool, well established in the area of pharmaceutical logistics	Additional (mobile) data entry client
ODK	Mobile data collection tool, can be easily configured based on MS XLS tools.	Additional (mobile) data entry client
RapidPro	Mobile data collection tool, using SMS and USSD	Additional (mobile) data entry client
OpenSRP	Open Smart Register Platform (OpenSRP) is an open source mobile health platform that allows frontline health workers to electronically register and track the health of their entire client population. Its data model is based on openMRS.	Additional (mobile) data entry client.

OpenLMIS	Enterprise-class, open source electronic logistics management information system (LMIS) purpose-built to manage health commodity supply chains	There is no obvious functional overlap. Potential learnings to be drawn from system architecture and community management activities.
OpenRBF	OpenRBF is a solution to facilitate the daily management of RBF schemes, such as data entry, strategic purchasing, and generation of payment order	OpenRBF started out as stand-alone product and then moved towards a close integration with DHIS2. OpenRBF deals with financial data since RBF schemes are part of Strategic Purchasing.

There are different ways to leverage existing systems. Below an approach is shown, where many different existing Health Systems are used for the purpose of data entry at facility level. This approach can help reduce the effort to rollout new software tools at the facility level. The schema also includes a software that serves as dedicated interoperability layer, as discussed as part of the openHIE concept. To reduce the number of APIs, external modules can send their data to a mapping layer before data is send on to the actual OpenIMIS database.

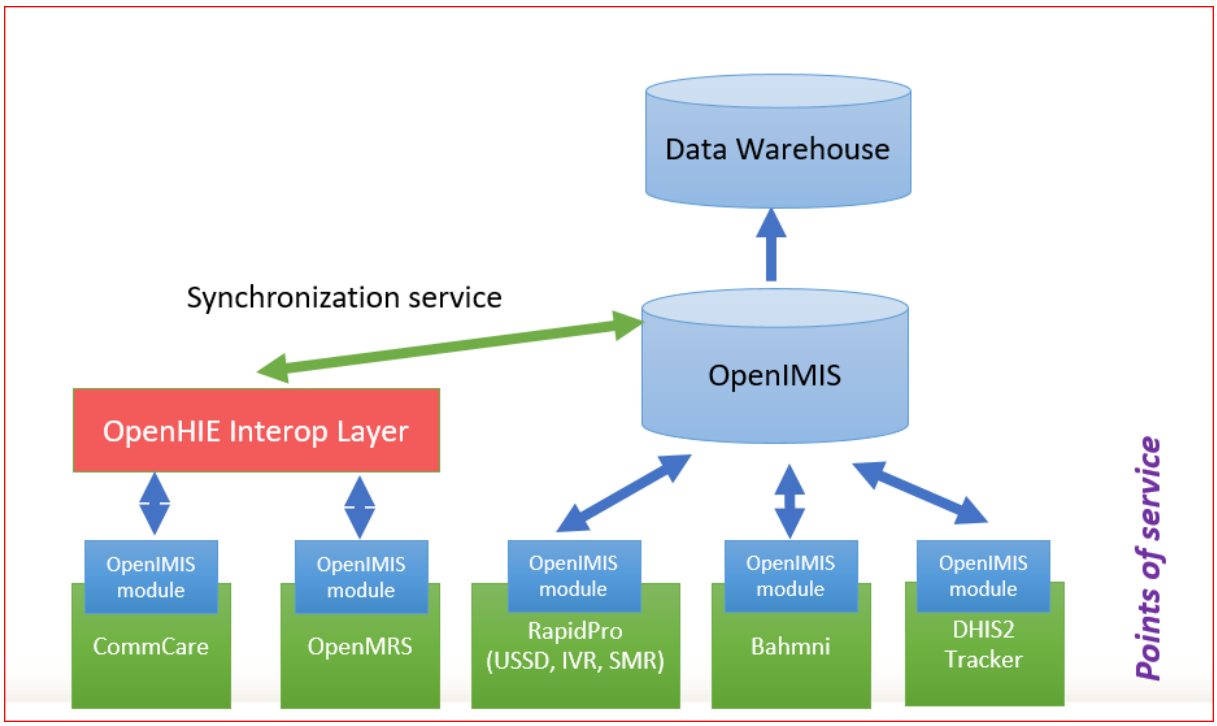

Figure 2: Re-using existing software solutions for facility data entry

3.6 Support to existing users/ IMIS implementations

One of the current main assets of the OpenIMIS program are the existing users in Tanzania, Nepal and Cameroun. In this technical roadmap, we want to identify the ideal and sustainable target architecture for an open source system that can support the development of a strong and stable user base and become the standard in as many target countries as possible. For each of the options, then user transition and data migration scenarios will be discussed, to ensure that the new system represents an attractive option for the existing users.

4 User requirements

Before designing the technical system architecture, it is necessary to look at the potential functional range the application needs to cover. For this procedure, we rely on already existing synthesized requirements from several projects, including JLN which have defined generic requirements identified through a multi-country collaborative process and the existing IMIS application, in use in three countries. This requirement overview serves to determine the software approach based on the following expectations:

- How can we build software that has a technological base allowing to accommodate all requirements in the long run?
- How can we accommodate the most urgent requirements quickly without losing the possibility to gradually expand the functional range?

4.1 Generic requirements

The most relevant collection of generic user requirements towards a Health Insurance System in low and middle income countries was created by JLN in between 2010 and 2012.
The table shows a summary of all processes identified as "common" amongst most Health Systems in Low/Middle income countries.

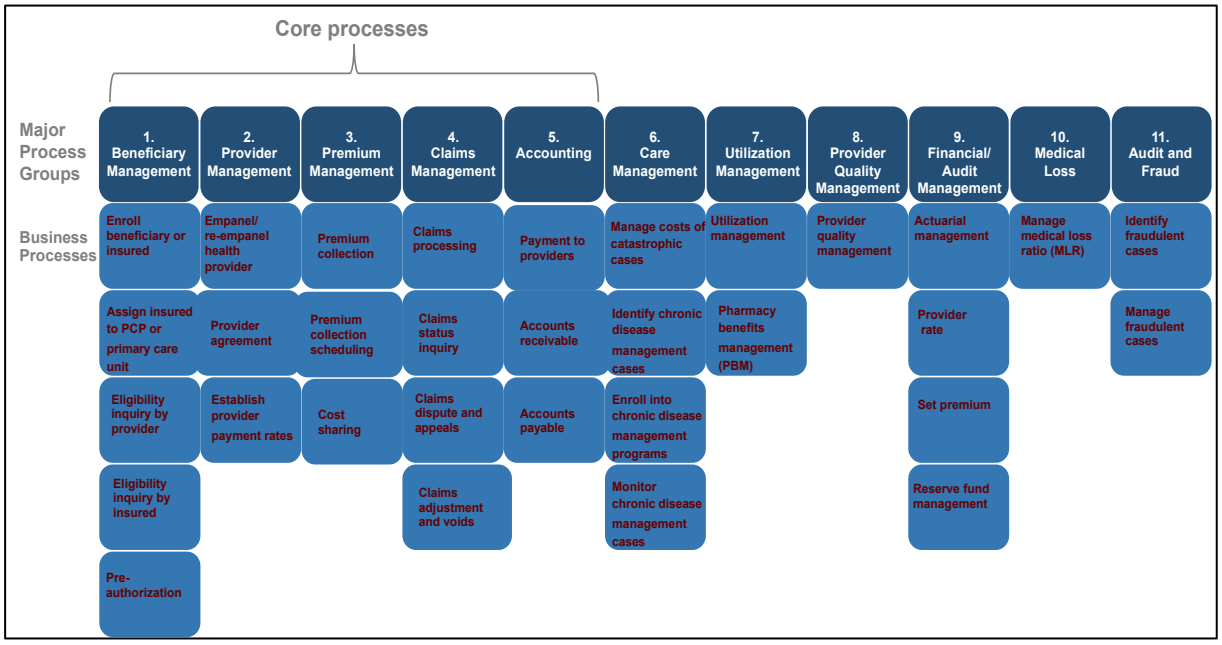

Figure 3: JLN CRDM Requirements

The requirements focus on the following four core procedures:

- **Beneficiary Management** – In this large definition, this includes all beneficiary interactions, including the interactions with the Insurance agency and providers
- **Provider Management** – This area focuses on interactions between the Payer (Insurance agency) and the care providers
- **Premium Management** – this covers a beneficiary interaction (Premium collection) and, administrative procedures at the Insurance agency
- **Claims Management** – This process ranges from claims transmission to acceptance/rejection

In addition, accounting was identified as a core function, although these functions will be typically handled in a separate application. The remaining processes are not being discussed in all detail in the JLN requirements documents and range from care management, which relates to care or coverage specific schemes (e.g. Catastrophic or chronic disease) to functions such as fraud management. The JLN requirements provide an excellent basis because they summarize the requirements from a range of countries, representing different regions, cultures and socio-economic situations (Ghana, India, Indonesia, Kenya, Malaysia, Mali, Nigeria, the Philippines, Thailand, and Vietnam). It is based on the Collaborative Requirements Development Methodology (CRDM).

However, there are also some limitations to using results from this approach for the OpenIMIS technical roadmap.

- The requirements were compiled in 2011, some recent trends may not have been fully considered.
- The approach aims to help countries build their own requirements based on these general requirements. Following this logic, the requirements can only serve as a first step towards the determination of OpenIMIS requirements.

4.2 IMIS functionality

The functions currently covered in the different IMIS versions serve as the major requirement basis. IMIS has successfully covered different use cases and its users constitute a quasi-community of practice. Also, it is an important option to offer the current customers the possibility to **migrate** to the future OpenIMIS version. IMIS functionalities will also serve to structure the modular analysis in the next chapter. The overview below shows a simplified modular view on the existing IMIS and the main user for each module, "X" representing the main use case, "(X)" a secondary use case.

Module	Facility	Insurance Agent	Insurance HQ
1. Reference data			X
2. Enrolment		X	
3. Registration, Pre-authorization	X		
4. Claiming	X	(X)	
5. Operational Agent Functions		X	X
6. Analytics functions	(X)	(X)	X

Table 1: Modules and actors

There are currently three relevant main different versions: Nepal, Tanzania and Master. Further development is planned for Tanzania in 2017. Below is a very general overview on main workflows and functions:

Module / Location Organisation	Workflow / functions
1. Reference data	• Configuration of reference data (Persons/ Families, Facilities, Payment methods, Services and goods, price lists, insurance products) • Definition at HQ level, transfer to mobile apps
2. Enrolment	• Two-step enrolment: Mobile Android app for pre-enrolment; Full enrolment at Insurance Office. Enrolment on paper forms with pre-printed QR-Code, Pre-printed card included. Card does not have a photo. Scan code, take picture, send via app to database • Contribution collection, recording (Mpesa Mobile payment planned for Tanzania)
3. Facility Enquiry	• Beneficiary goes to Facility. Shows card. Facility uses app to **identify** via QR code or number (Ideally online, but also possible offline) • **Pre-authorization** – checks in online DB for eligibility / remaining credit • Access through mobile app or browser
4. **Claims submission**	• Encounter details can be recorded on paper form, then entered for claiming into mobile app or browser (by facility or Insurance agent/District Office)

	• IDs: Facility Code, User code, Patient, Claim Form Number, several diagnosis codes, Serval service codes (drop down list), Emergency/Referral
5. Head Office Operations	• Claims management (Verification, Approval, Transfer for Payment) • Operator Office (HQ, some functions also at District Offices)
6. Analytics	• Standard reports (Transaction lists, aggregations) • Available in system directly and in external MS BI tool.

Table 2: Modules and functions (Simplified structure based on IMIS Master documentation)

In addition, the IMIS application has an API to a web based SMS service for communication with Insurance Agents. Also CRM functions are planned (direct messaging to Beneficiary - USSD, SMS).
An up-to-date list of planned future functions can be found in the JIRA Issue queue https://openimis.atlassian.net/projects/OP/issues/OP-20?filter=allopenissues , more general information on the strategic direction of the functional roadmap can be found in the Wiki https://openimis.atlassian.net/wiki/spaces/OP/overview .

4.3 Requirements summary

Based on the discussions above, we are confident that the existing JLN requirements and the functions build into MS IMIS can serve as a sufficient foundation for defining the system architecture and approach of OpenIMIS.

5 Technical analysis of MS IMIS

All versions of IMIS (country specific versions and the Master Version) were developed by Swiss TPH in cooperation with the Tanzanian software company Exact. The Nepal version was then modified by the local GIZ team. All countries (Tanzania, Nepal, Cameroon and now also Congo-South Kivu as the first French IMIS implementation) received free IMIS licence from SDC. Some of the outputs were also the thorough technical and functional documentation. This analysis serves to identify how this investment can best be leveraged for the development of an open source OpenIMIS.

The value of the investment is not only represented by the software itself. Beyond the software version available for implementation, further assets have been generated, such as:

- Consolidated multi-country requirements that have been transformed into software functionality, which have been tested in different settings and documented
- Data model and analytics structure, technical documentation and test cases
- Implementation experiences: Time plans, resource needs and evaluation reports for implementation and maintenance.

5.1 System documentation

A good documentation is an important basis for system deployments in different contexts. We are currently aware of the following documents:

Type	Documents	Version
Specification	In 2012, software specifications were documented, including use cases. These specifications represented the original expectations towards the software. The documents have been updated for each country. Request for changes are referenced to these specifications. • **2012-07 IMIS_Specification_Phase_1_v4.doc** • **2012-07 IMIS_Specification_Phase_2_v2.doc** A separate specification was created for the Analytics and Reporting system: • **2016-03 HPSS_IMIS_Specification_Analytics_v1.6.doc** The original document naming divides the software specification into phases (which correspond to modules).	2012
Specification	In 2017, a specification document of the consolidated master version was created: • **HPSS IMIS Specification Master version_3. 3.5.pdf** The document covers all aspects of the maser version, including detailed overviews and data fields for the three country versions. In section 4.4 “Additional functionality” the Request for changes are lists, that were the basis for the integration of country functions into the master version.	2017-01
Technical documentation	The main technical documentation consists of a descriptions of the system architecture, programming approach, standards and methods, the menu structure and interface flow, functional area descriptions and database design. Furthermore, web services and the mobile phone apps are described. • **Technical Document 24042017.pdf**	24 April 2017 Version 17.3.11
Documents not available	• Release history: Including all detailed requests for changes (RfC**)** • Compilation & installation guide: Including Server & DB Setup • Test cases • Implementation experiences: Time plans, resource needs and evaluation reports for implementation and maintenance. This material does not exist in an updated version or has not been made available for analysis.	

Table 3: Documentation

As of beginning of 2018, this documentation is transferred step-by-step to publically available documentation repositories such as http://openimis.readthedocs.io/ . Up-to-date references to the documentation can be found on https://openimis.org .

5.2 Technical analysis

The technical analysis serves to describe the current system and to discuss aspects that help evaluate how sustainable the current system approach is. The analysis is based on system documentation, additional information from SwissTPH and feedback from GIZ Nepal.

Item	Description of IMIS features	Comments on sustainability of approach
1. Programming languages	The programming language is VB.NET (Microsoft Visual Studio 2012). VB and C# are the two major programming languages used in MS.NET. Since 2017, Microsoft seems to prioritise C# over VB.Net, raising fears that support for VB.NET will be eventually phased out.[2]	In many developing countries Microsoft is still the most widely used developing environment. Especially VB is widely used amongst developers.
2. DB Version, OS Version	• Mobile app: Android 2.3 • Minimum: MS Server 2008, MS SQL Server 2008 MS SQL Server 2014 is required for running parts of Analytics and Reporting tool.	
3. Licenses	The application can only run on a Microsoft (Server) Operating System, which requires a license. For the database, there is the choice between the MS SQL Server Express or the full version. Countries can start with the free Express version, once usage grows it is useful to switch to the commercial version (As Nepal and Tanzania have done). The approximate licences cost is around 4000 EUR.	License cost is low taking into account the scope of most projects. Small implementations can be realized license free
4. Performance / Scalability	The current countries have relatively small member basis and claiming numbers compared to other countries.[3] Tanzania has witnessed no performance issues (with exception of running long extracts for off-line installations). Nepal witnessed challenges while the full ICD code list (ca. 12.000) was loaded as list of value in 4 fields. The problem disappeared with the reduction of codes.	In general, the application seems fit for volume scaling. Some performance tuning may be useful in association with national roll-outs in larger countries.
5. Extensibility and parametrization options	IMIS allows to parametrize options, such as enrolment and claiming workflows, configuration of insurees, products, prices and policies. Additional parameters can be changed at the database level. The business logic and database layer of all country specific versions are identical, differing only at the presentation layer by well-defined features (labels of data fields, drop down lists for data fields, the second language of user interface, displaying of specific data	IMIS covers a wide range of the domain specific functionality that should respond to many future requirements. In case a country wants to include unforeseen functions, it should normally request the centralized development team to make the changes and wait for the changes to be released (to the country or as part of the general master). Countries

[2] https://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Visual_Basic_.NET#Language_history - "The primary reason for the decision is that with time VB has become less popular with professional developers relative to C#."

[3] NHIF Kenya, 2016:
- 6 Mio Members, Monthly 60.000 new members (10 Mio Dependants, 200.000 new monthly)
- Outpatient visits > 2 Mio, monthly 110.000, 5 Mio Impatient admissions (60.000 per month)
- > 17 Mio Contribution records, 300.000 monthly
- Individual accounts upload: > 140 Mio, 3 Mio monthly

Philippines (2015): 93 Mio Beneficiaries (40 Mio Members, 53 Mio Dependents)

	fields, mandatory or optional entry of data into specific fields). Super users can modify data entry screens by excluding a field, but not by adding. The core is maintained centrally and development of diverging country versions is not desired. Functional enhancement based on core functionality should be built into core after approving requirements and testing compatibility with existing functions. This approach is chosen to control the complex transactional business logic and data integrity. The Master Version has some predefined exit points for incorporating the so-called escape procedures for modifying business logic of IMIS, e.g. verification of insurance numbers, communication with SMS portals and mail servers. The scope of exit points can be extended. In addition in Tanzania the development of an API is underway for the enrolment data collection process	attempting to change functionality themselves need to create a separate version and have upgrade problems. The possibilities for countries to add functions to the core functionality by themselves, e.g. through additional modules or APIs are thus limited.
6. Internationalization	Internationalization features allow a system to adapt to local requirements. • IMIS allows to maintain two languages via resource sheets • IMIS currently does not have a repository of all national calendars, in order to support Nepali date of Birth, a workaround at the presentation layer is in place (also due to budget reasons). • Typical requirements additional for internationalization are currencies, numeral systems, time zones, regulatory requirements, name structures, national IDs, ZIP codes, …[4]	IMIS was first developed for a project in Tanzania and later adapted to different national settings. It relies on standard internationalization features of Microsoft tools, which cover a wide range of typical requirements. An analysis of SwissTPH shows that IMIS should be prepared to handle most future requirements regarding currencies, time zones or languages.
7. Data import and interoperability	Claims and enrolment data can be imported via XML files. This method is used by the Android apps to push data to the MS file server. For inserting large administrative data lists (e. g. territorial/administrative units, medical items/services) SQL scripts are used. IMIS can export analytics data to a DWH via an ETL. Also payment data can be exported to accounting system to initiate payment processing. In Nepal this is done through custom made SQL report.	The current IMIS master version does not provide a far-reaching API, which would allow other systems to submit or receive all data after authentication (e.g. like DHIS2 does via a RESTful Web API, where practically all data is covered, including metadata such as facility lists). API development has been initiated, there is now a stateless API under development (Tanzania).
8. Database abstraction	The system architecture has a special database tier. This layer is used to connect application and database.	As of our understanding, it is currently not possible to connect open source databases (because of MS restrictions), leaving as only (hypothetical option) to replace the MS database with an Oracle database.
9. Security concept	IMIS has usual security functions (user roles, auditable log files, historical records, session expiry after a defined time with automatic logout,)	Microsoft applications and databases normally can provide a high level of security. Database vulnerabilities are quickly fixed. The data security will depend on how the application is implemented.
10. Browser access	The user interfaces of the main application are well displayed on typical PC or laptop screens (e.g. Width of 1024). The screen design is not mobile responsive in the sense that it does not adapt its layout to smaller screens (tablets, mobile phones). Mobile access takes place via Android apps.	Mobile responsiveness will become more important since the diversity of user devices will continue to increase, especially in different settings (facilities size, use case, countries, etc.).

[4] https://en.wikipedia.org/wiki/Internationalization_and_localization

	The screens have a typical professional Microsoft look. Countries can adapt the interface, by omitting fields that should not be displayed.	

Table 4: Elements of system analysis

5.3 Country feedback

In May 2017, the GIZ consultants visited the Nepal GIZ team to get a better view of the state of the IMIS implementation in Nepal. The general feedback was that Nepal is satisfied with the overall functioning of the application. However there are also some areas where GIZ Nepal feels that the application does not adequately cover all needs.

Nepal has been modifying the application locally, which makes it more difficult to upgrade to the next version. Nepal has not yet fully tested or implemented the Master version.

Nepal has made some enhancements to the application that might also be interesting for other countries. For example, on the start page a Dashboard built on Google Graphs has been integrated, built on a Web API.[5] With the current system architecture, such enhancements are more difficult to share between countries and difficult to migrate in case of version updates.

5.4 Conclusion

Over the last years has IMIS developed into a flexible health insurance solution that is ready for deployment and should be able to accommodate most immediate needs of many potential users. But as discussed, IMIS also has some limitations regarding parametrization, modularity and interoperability. Therefore we believe it is useful at this stage to compare the continuation of IMIS development against other options.

[5] It should be noted that it is also possible (and maybe more convenient) to use an external app such as DHIS2 to build such dashboards.

6 Main system architecture options

As part of this analysis, three major approaches have emerged, which are roughly outlined below:

No.	Approach	Main points
1	Open Source the existing MS IMIS Master and continue its development	• In the short term, it is the most attractive option to build on this working system. • Yet it needs to be determined whether the long-term technical constraints can be overcome. • In any case a transition strategy for existing MS IMIS users needs to be defined.
2	Rewrite MS IMIS into OpenIMIS, using open source code and tools	• Starting an application from scratch allows to define and apply best practices approaches for modularity, interoperability and usage of existing standards • As with any entirely new application, the starting phase will require a lot of testing and error corrections, especially when trying to apply innovative concepts.
3	Build on existing platforms	• OpenHIE offers a growing ecosystem of interrelated components • DHIS2 offers a well-supported software platform with a strong community • Connecting existing systems may result in a technically more demanding solution and will require a clear approach to manage external dependencies

Table 5: Principal software architecture and technology options

These three options represent simplified project types which should be combined. For clarity reasons, we are first going to present the "pure" approaches.

6.1 Option 1: Continue MS IMIS

Option one (1) suggests to actively continue development of the MS IMIS, maintaining the Microsoft based approach. The obvious advantage of this approach is to build on an existing, stable software, which would make development activities and required resources more predictable. Although VB.Net is seeing a slight degrade against C# as a programming language, it would still be a solid basis for further development for the next years.

The graphic below outlines the system architecture, which would remain unchanged in its structure, only adding additional existing apps, such as DHIS2 as Data Warehouse and potentially other data entry apps.

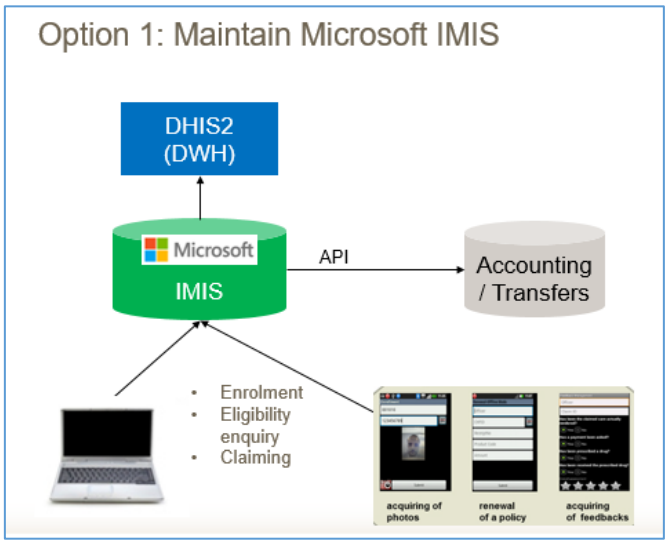

Figure 4: Option 1: Maintain MS IMIS

However, as shown in the technical analysis, there are also some conceptual challenges to transform MS IMIS into a more sustainable solution. The development could focus on areas such as interoperability,

implementing standard methods such as RESTful/ASP.NET Web API to enlarge the data exchange options. It would also be useful to allow for code contributions to core functions via apps or visualization of external apps in browser. Also, scalability would have to be tested to make sure the application is fit for larger countries with bigger member groups and higher claiming volume.

In most scenarios, the licensing cost can be contained to around 5.000 Euro per country, which in most scenarios only represents a neglectable fraction of required overall investment.

More important regarding costs is that the initial implementation efforts seem[6] to have been quite high for some of the countries. It is not clear which part of this effort is due to the earlier immaturities of the software, the country-specific deployment situation and country-specific requirements. Still, as part of the next steps, one activity could be to systematically evaluate implementation experiences and develop technical and organisational approaches to make country customization and implementation smoother and less costly.

6.2 Option 2: Rewrite OpenIMIS

The second approach would rely on the functionality of MS IMIS but transfer it into an open source environment, establishing an approach without license costs.
At first glance, this looks like a quite predictable and low-risk technical effort. The application code and database scheme can be converted or re-written from scratch respecting functionality, database structure and business rules.

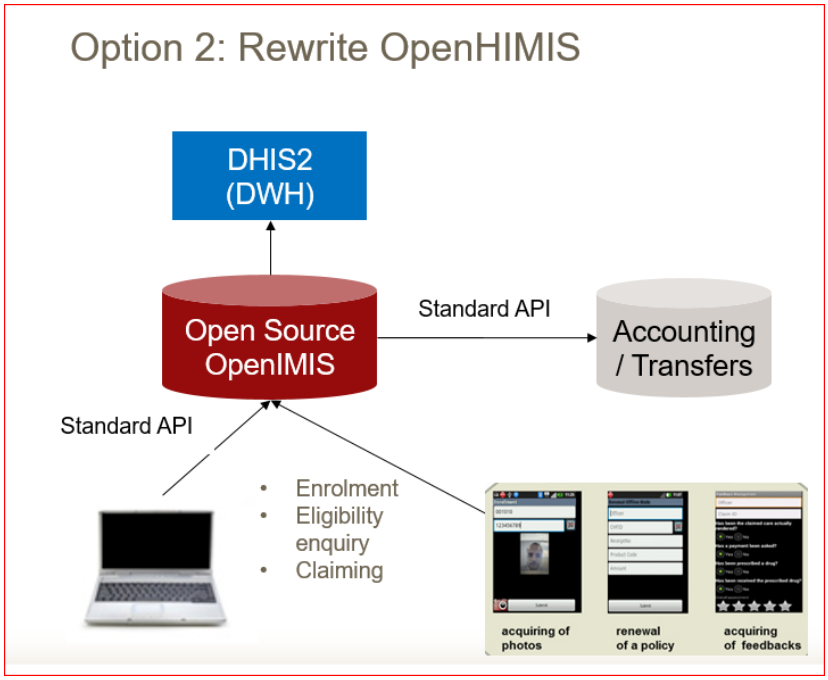

Figure 5: Option 2: Covert MS IMIS to Open Source Technology

However, in order to have a lasting impact and justify the investment, the rewrite should go beyond simply reproducing the current system. Some of the conceptual challenges to be tackled are to produce

[6] As of now, we do not have detailed implementation effort information from projects. However some observation point toward these high efforts.

a software that fully corresponds to the concepts of an international standard software, allowing for a full parametrization of functionalities, including all typical language, calendar, currency requirements. In addition, standard interoperability approaches would be integrated, while modularizing the design.

This would also allow to apply internationally established data standards, such as HL7 FHIR or frameworks such as OpenHIE. HL7 already provides well defined data models, libraries and APIs for some important data objects, such as basic patient data (client demographics), health facility data (clinical and administrative), patient records, claims, invoices, etc. This would allow to build a **micro-service** oriented architecture, where each of these services have their own database structure, offering the possibility to develop the micro-services more independent of each other. This independency can enable development partners or countries to develop different versions or functional enhancements for specific use cases, without interfering with the rest of the core software.

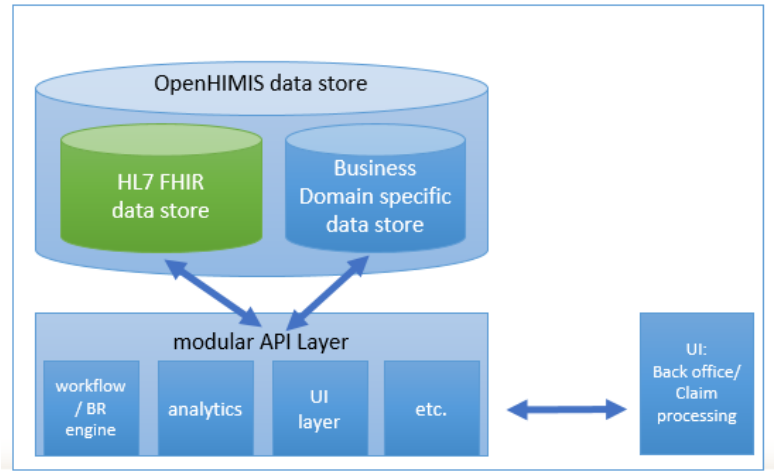

Figure 6: Micro-service based implementation of HL7 FHIR data model

The HL7 approach would allow to be compatible with many other health information initiatives, including many EMR solutions and solutions of private health insurance providers, that OpenIMIS may have to interact with. However, this would also require further exploration. While HL7 is gaining importance in the sphere of large commercial EMR and medical device makers, it needs to be evaluated whether it is mature enough to be used as a central concept of OpenIMIS in the context of low-resource countries.

The initial investment would certainly be higher than with the MS IMIS solution, and getting the solution to the stage of maturity and stability where MS IMIS is now, will probably 1-2 years. The more innovation is applied as part of this process, the higher the potential rewards, but also the uncertainties regarding cost, time and software quality.

6.3 Option 3: Build OpenIMIS on existing platforms

A fundamentally different approach would be to build on the basis of an existing framework or platform. As discussed in the section on software principles, in the health software space there are several widely known solutions that are fit for many related use scenarios in low resource settings. **OpenMRS** is a medical record system that is mainly used at facility level, but also supports some national scale programs. **DHIS2** started out as a national statistics system for aggregate health service data (Health Management Information System (HMIS)). Recently it has gained considerable traction and functionality related to patient tracking, gradually integrating the basis for transactional approaches.

Both systems offer common advantages over a software development that starts with a proper codebase, both in the area of community access and technical synergies. Comparing OpenMRS against DHIS2, we found DHIS2 to be more fitting for health insurance purposes, mainly due to its strong Data Warehouse, its flexible organisational hierarchies, its framework approach, and the role that it is assuming for many centralized government agencies. Also, it has a well-established training and support structure, that has allowed it to act as an anchor point for many national health information architectures.

For DHIS2, some initial architecture conceptualization was done, as shown in the graphic below.

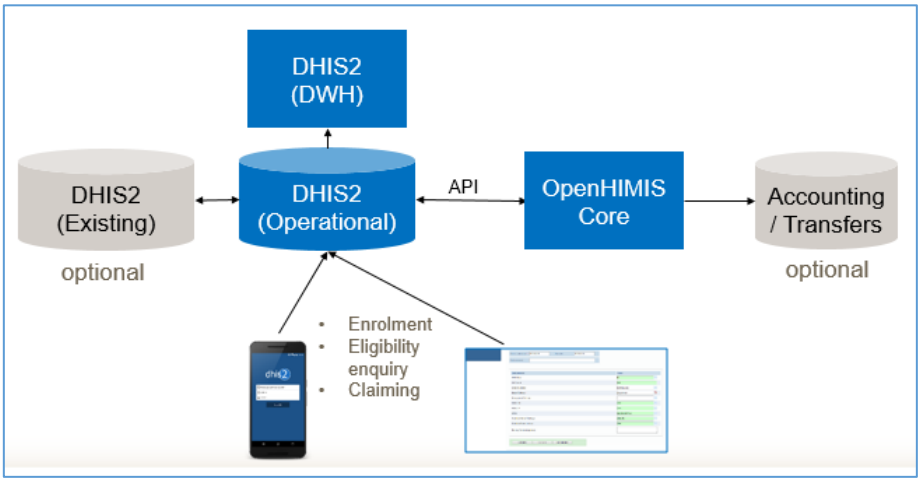

Figure 7: Option 3: OpenIMIS based on DHIS2

Eventually, this option was discarded because it appeared to be difficult to align the openIMIS objectives with the development path of the DHIS2 core.
Therefore discussions focused more on a closer cooperation with partners of the openHIE framework. These modules cannot replace the current openIMIS, but they can take lower the development burden for specific functions that are already available in existing modules of the openHIE ecosphere.

7 Way forward

Discussions between GIZ, SDC and Swiss TPH in July and August 2017 pointed towards a step-wise integration of modularity, interoperability and open source components. During the technical workshop in February 2018, these elements were discussed and agreed upon by a wider team of stakeholders.

An important component of the discussions were technical suggestions made by SwissTPH. They are summarized below, complemented by some of the major comments given by GIZ.

Summarized suggestion by SwissTPH	Comment
1. OpenIMIS should be developed based on the current IMIS, using MS VBA.	Agreed in the sense that for the time being all existing core functions should remain in Microsoft Technology. For new functions it should be analysed case-by-case whether new elements can be used (micro services, open source, existing libraries), thereby stepwise modularizing the core.
2. IMIS will be modified in such a way that it will support besides MS SQL Server **database** also other databases (e.g. open source or license free database). This will also allow running on Linux OS and gives partners technology choices. Migration tools will be prepared.	Agreed, based on specific customer demands. *(This has a lower priority)*
3. The data warehouse of IMIS will be integrated with alternate front–end besides the currently used front-end MS Excel. Priority would be DHIS2, but it should also be possible for other reporting/DWH/BI products.	Agreed. Pro-actively pursuing DHIS2 integration corresponds to requests of different partners. Still, vendor-independent approach is important.
4. The current **RESTful API** layer will be completed (and made fully compatible with the relevant standards) in order to facilitate full replacement of web form functionality in case it is needed for ensuring online communication with adjacent software systems.	Agreed. It has been a new information for us, that IMIS uses RESTful API.
5. A **data format** in conformance with HL7 FHIR will be provided for data exchange on claims as an alternative to the current native XML based format.	Agreed. In a later step, it can be analyzed how micro-service based FHIR libraries could also contribute to long-term core modularization.
6. Data formats based on CSV and/or Excel will be offered for initial loading of selected **registers** (it is already available but only for some registers and it is not published yet).	Agreed. This will speed up implementation and help transfer maintenance responsibility to countries. Also continuous data exchange with specialized registers, such as facility lists, can be explored in line with openHIE approaches.
7. The existing mechanism of **exit points and escape procedures** will be enriched to allow country specific modification of more aspects of the business logic in case an explicit configuration on the user level is not advantageous for a specific case.	Agreed. This seems like a promising approach to allow countries to inject further functionality. Since this can deeply affect core stability, the procedures should be well structured and documented. *(This has a lower priority)*

During the technical workshop in February 2018, an approach was discussed and agreed upon, that aims to combine elements of the three options discussed, as shown in the overview below:

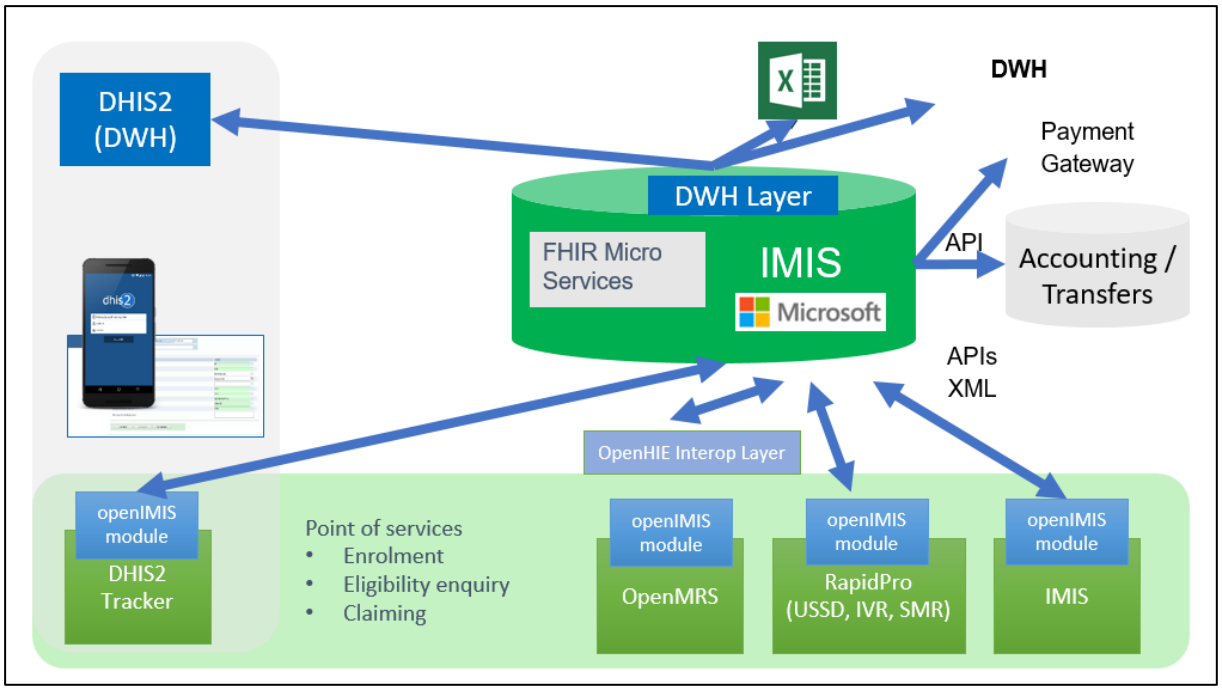

Figure 8: Synergistic system approach

The major elements from the three different options are included in this approach:

- **Option 1:** For the time being, the current MS VBA code based application will be maintained. However, investment into VBA code will be limited. A special focus will be on quickly developing further interoperability features of openIMIS, opening up the legacy system to modular enhancements. This is based on the assumption that the current IMIS core will not need to be replaced in the coming 2-3 years. To confirm this assumption, some more analysis would be needed to be certain that the core holds up to the expectations regarding interoperability, modularity and accessibility to an open source community.
- **Option 2:** For all major functional updates, it will be verified whether this can be done based on open source technology or existing HL7 standards or modules. A pre-condition of that is to modularize IMIS core components step-by-step. A complete replacement of the current MS IMIS core will not be aimed for during the next two years but rather at a later stage.
- **Option 3:** Integration with elements of the openHIE architecture will offer important opportunities for re-using existing technology, enable strategic partnering with other software communities, and achieving more sustainable funding. We believe that modular technology allows to pursue these synergies, at the same time respecting data security, privacy aspects and other organisational boundaries.

A major aspect of the **investment case** is to preserve the current customer base. We believe we should be offering options to upgrade the existing systems and to continue strengthening it by integrating it further with the national **health system architecture**. We believe that actively integrating OpenIMIS with established opens source systems, can broaden the future customer base. We also acknowledge that the **current IMIS core functionality** (especially the management of policies, beneficiary contributions and transactional processing of claims) represents one of the major assets of IMIS and can also be of a future OpenIMIS.