Introduction

Traditionally, most insurers predict claim payments by taking similar groups of policies and analyzing their historical trends such as frequency of claims, and amounts paid toward such claims. Based on patterns obtained from the datasets, new estimates of claim reserves are manually produced as depicted in Figure 1 below:

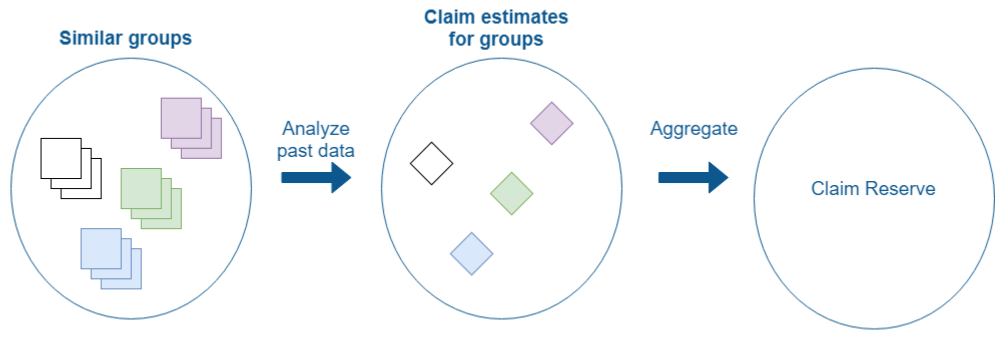


Figure 1: Paper-based insurance claims processing and adjudication workflow

This method produces a good estimate of Claims Reserve, but the **final claim payment is always somewhat different** than the expected one. There are two sources of the difference:

1. Actuaries take the group of similar policies and do not take into account individual characteristics of policyholders. Thus, it can turn out that some individuals are not following the patterns of a group and claims are much smaller or higher for them, impacting the aggregated result.
2. Between a claim's initial filing and full payment, the final amount of the claim can change drastically thus an insurer has set up additional outstanding reserve for possible losses.

Today, Artificial Intelligence (AI) techniques such as Machine Learning (ML) can be used to discover patterns from non-linear datasets (Figure 2).

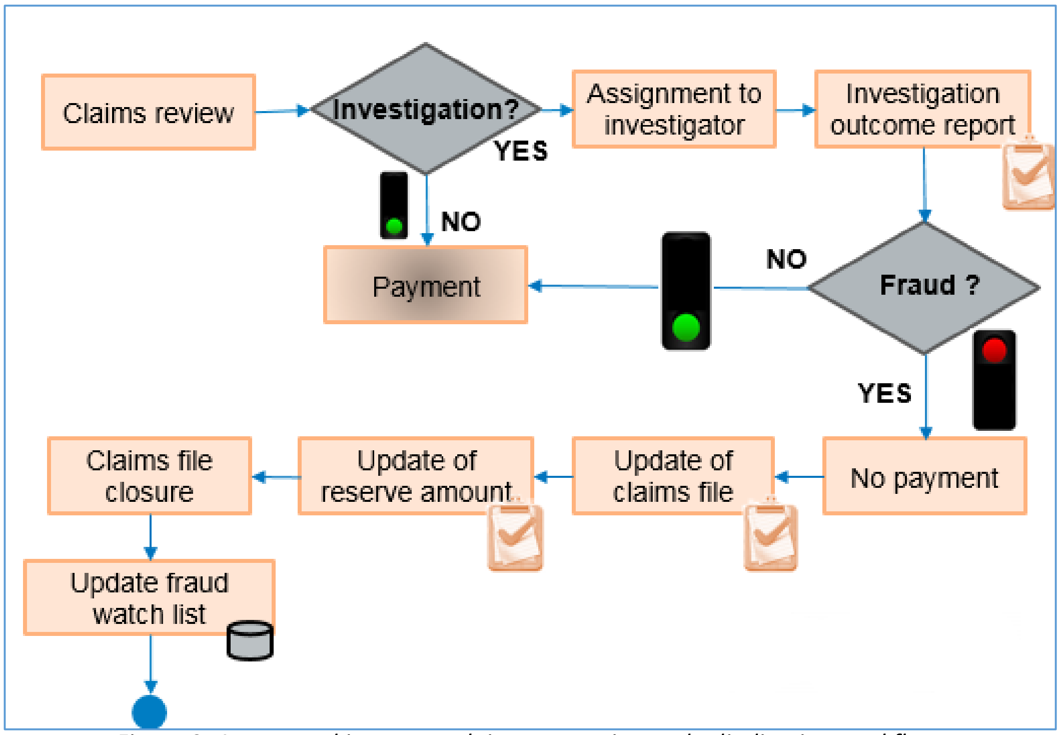


Figure 2: Automated insurance claims processing and adjudication workflow

Based on the learned pattern, the algorithms can predict payments on a claim; or even identify potential fraudulent claims and flag them for investigation. Insurers now have the option of achieving far better claims management by utilizing the technology in the following ways:

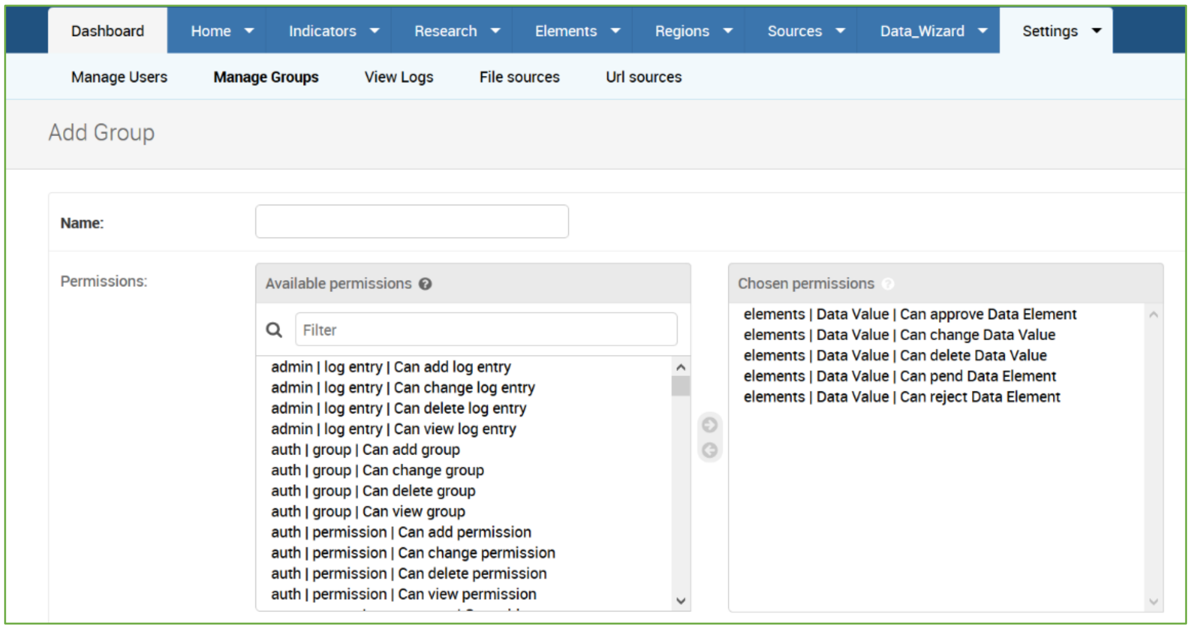
1. Use rule-based engine to pre-assess claims while automating the evaluation process.
2. Use artificial intelligence to automate claims fraud detection through rich data analytics.
3. Use machine learning to predict patterns of claim volume.
4. Augment loss analysis.

Improved Claim Review

The Improved Claim Review requirement describes two distinct mechanisms aiming at Claim processing automation. To enhance openIMIS functionality, our specific assignment will be to improve the functionality of the new platform by providing Python-based **Configurable Claim Review Engine;** and **AI-based Claim Automated Adjudication and fraud detection**

Work Package 1: Build Configurable Claim Review Engine

A **Configurable Claim Review Engine** can be used at Claim entry or submit level to validate a claim prior to any further treatment. The new openIMIS architecture already integrates Django-rules for access management. We will use Django Framework capability to build rule-engine claim validation engine as an extension to current claim module. The Django module will allow countries to dynamically change the validations performed (via rules) when a claim transits from entered to submitted as shown in one of our application developed in Django



Add Group

Name:

Permissions:

Available permissions

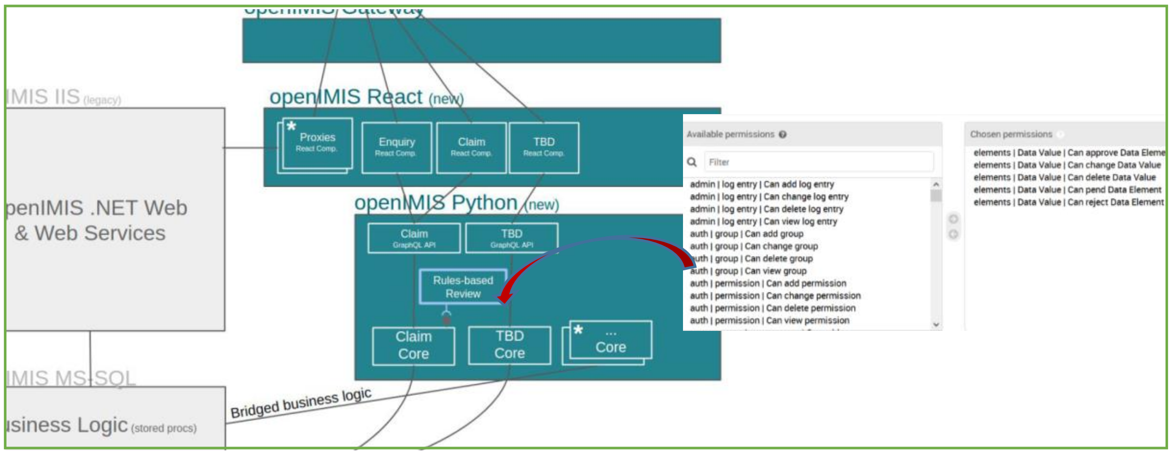
admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | permission | Can view permission

Chosen permissions

elements | Data Value | Can approve Data Element
elements | Data Value | Can change Data Value
elements | Data Value | Can delete Data Value
elements | Data Value | Can pend Data Element
elements | Data Value | Can reject Data Element

Figure 3: Django rule-based context implemented in one of our solutions

For example, if the rule requires that the insuree gender, date of birth, the health facility type, available such rules will be moved from the left box to such **(available rules)** to the right box **(chosen rules).** All the datasets loaded from database into the claim core must be evaluated against these chosen rules. The available rules will be created and configured by (admin) users and tested against established data governance framework.



admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | permission | Can view permission

elements | Data Value | Can approve Data Eleme
elements | Data Value | Can change Data Value
elements | Data Value | Can delete Data Value
elements | Data Value | Can pend Data Element
elements | Data Value | Can reject Data Element

Figure 4: How Django rule-based context will be applied in claims processing

Work package 2: Build AI-based Claim Adjudication and fraud detection

Rule-based algorithms use relational rules to describe data. However, as systems become operationalized, a rule-based approach to machine learning can become very complex. For example, it is likely that hundreds of exemptions to 100 predefined rules might emerge to register incoming data. Artificial Intelligence (AI) technologies can be seen as an extension of the rule-based mechanisms. In claims automation, AI-based technologies such as pattern recognition and machine learning will be used to will support the following:

1. Fraud detection: Machine learning models can be used to detect possible fraudulent claims hence save on revenue and corporate image;
2. Policy compliance: Provide flexible techniques for calculating coverage and payment for each claim according to set insurance policies;
3. Decision making: AI eliminates much of the guesswork associated with decision-making; thus decisions become more accurate, correct, and consistent;
4. Time and cost saving: Drastically reduce processing time between insurance and healthcare provider systems.

Fraud detection

Insurance companies lose an estimated **US$30 billion**[1] a year to fraudulent claims. In fact, as more and more customers use online services, the potential for fraud has increased dramatically. To mitigate this risk, machine learning may be used to identify potential fraudulent claims faster and more accurately, and flag them for investigation.

Linear techniques, neural networks, and deep learning are used together in order to spot fraudulent behavior. Because rule based and linear regression models cannot detect advanced fraudulent behaviour, we will adopt a holistic approach to implementing our solution as depicted in the figure below:

[1] *Enabling insurers to become AI-driven enterprises powered by automated machine learning: Accenture (www.accenture.com.)*

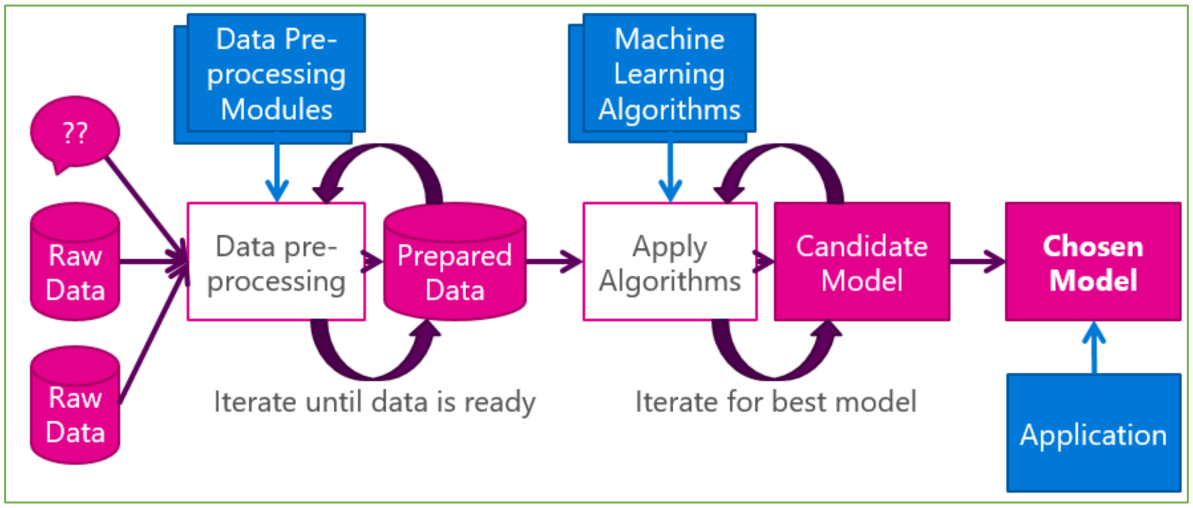


Figure 5: Methodology for implementing AI-based Claim Adjudication and fraud detection

The motivation behind this iterative approach is that deployment of ML module in openIMIS will focus on building the proper machine learning pipeline required for prediction. The following is a brief explanation of each of the stages of the above workflow:

1. **Identify raw data:** Identifying the relevant data sources will be the first step in the cycle. The raw data may come from health facilities or insurance takers in the formal sector.
2. **Pre-process data:** The will be used to make sure that the data received from source systems is clean, secured, and governed. Tools such as Talend Open Studio will be used for data Extraction Transformation and Loading (ETL) into a database
3. **Apply machine learning algorithm:** Based on the nature of the prepared data, we will apply several machine learning algorithms on the training and test datasets. The candidate algorithm will be from supervised ML models such as Naive Bayes, Linear Regression, Random Forest, and support vector machines). Candidates for unsupervised learning include K-Means clustering and Hidden Markov Model
4. **Iterate and Train:** To identify the most optimal model, iterative training will be done depending on the type of data and algorithm. The training process may be supervised, unsupervised, or reinforcement learning to find the best performing algorithm.
5. **Deploy and Maintain Chosen Model:** The optimal machine learning algorithms will be used to create the automated adjudication and fraud detection module that will be integrated into openIMIS. After deployment, we will continuously review the model to evaluate its capability of making predictions based on new incoming data.

Work Plan for Claims Automation

Proper planning is crucial to identifying and mapping the project activities, and resources and scope the system requirements specification. Figure 6 shows a breakdown of high-level tasks and activities that will be undertaken to realize the objective of automating claims processing and adjudication:

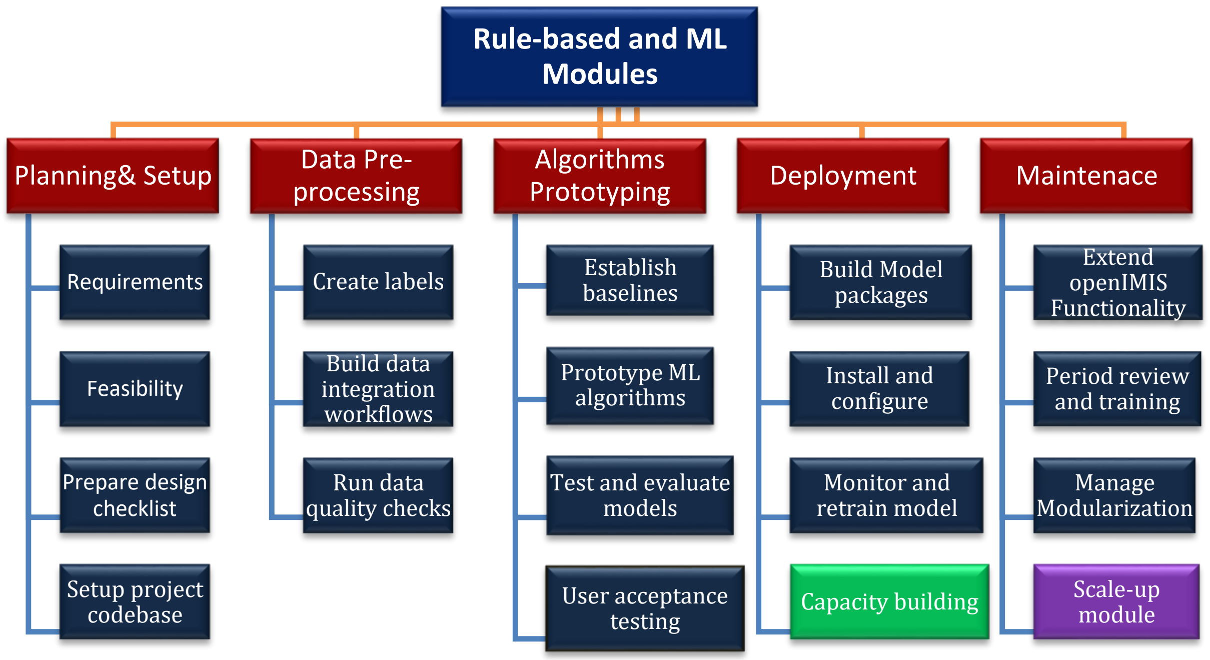


Figure 6: Work breakdown structure for automated Claims adjudication and fraud detection

The following is a breakdown of each of the high-level project activities depicted in the WBS illustrated above:

1. **Planning and project setup.** This the most fundamental stage in the lifecycle of the rule-based and AI module development that will entail the following sub-activities:
   1.1. Define the task and scope out requirements
   1.2. Determine project feasibility
   1.3. Prepare design checklists
   1.4. Set up project codebase

2. **Data Pre-Processing.** This is a crucial step that will be used to prepare and label data mined from openIMIS and other sub-systems. The data will be used for training and testing the ML models. Sub-activities include:
   2.1. Define ground truth and create labeling documentation
   2.2. Build data integration workflows and pipelines
   2.3. Use validation tools to check quality of data received from source systems

3. **Prototyping and Validation**. This is the most demanding activity broken down into the following sub-activities:
   3.1. Establish baselines for model performance
   3.2. Implement a simple ML model using initial data pipeline and continuously overfit the model to training data. The following is a sample Python codebase of the ML module to be implemented within a Docker container or virtual environment:

```
data/
docker/
api/
  app.py
project_name/
  networks/
    resnet.py
    densenet.py
```

```
models/
  base.py
  simple_baseline.py
  cnn.py
configs/
```

3.3. Test and evaluate model on test distribution to understand the differences between train and test set distributions as shown in Figure7.
3.4. Refine the model to ensure it provides desirable results and downstream user behavior
3.5. Conduct User Acceptance testing on the alpha and beta versions of the prototype

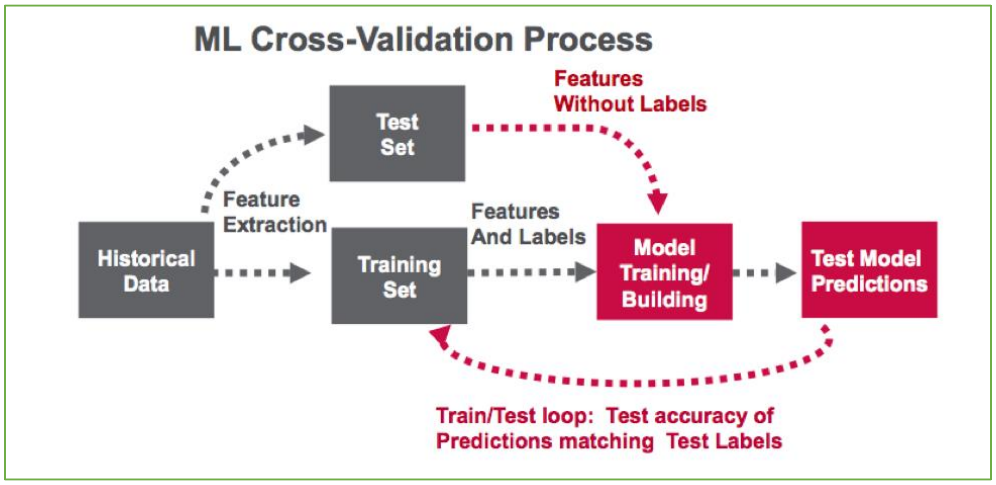
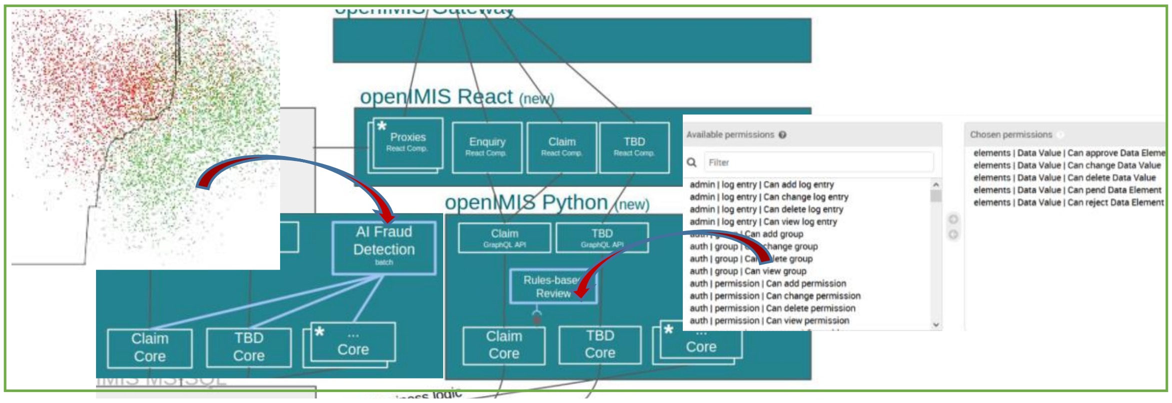


Figure 7: ML model testing and evaluation

4. **Model deployment and maintenance.** The chosen model will be integrated into openIMIS for deployment and utilization. The following are sub-activities of this work package:
   4.1. Package the module into a Docker container and expose a REST API for inference
   4.2. Install and configure the new module for limited number of users before rolling out to scale
   4.3. Monitor live data and model prediction distributions for possible adjustments
   4.4. Periodically retrain model to prevent model staleness
   4.5. Provide capacity building and knowledge transfer to users for ownership, and better utilization of the new openIMIS prediction features.

The expected deliverables for the Rule-based and AI-based claims automation work packages include: incremental rule-based and ML prototypes; solution specification document; architectural and ML Models; metadata dictionaries; workspace configuration tools; and REST API for integration with other sub-systems connected to openIMIS. Figure 8 shows the overall conceptual view of our solution:



admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | permission | Can view permission

elements | Data Value | Can approve Data Eleme
elements | Data Value | Can change Data Value
elements | Data Value | Can delete Data Value
elements | Data Value | Can pend Data Element
elements | Data Value | Can reject Data Element

Figure 8: Conceptual view of rule-based and AI Claims processing module to be integrated into openIMIS