Blockchain · Computer Science · Industry

# Automating Healthcare Claims with AI and Blockchain: A Transformer-Based Smart Contract Framework

2025

# Automating Healthcare Claims with AI and Blockchain: A Transformer-Based Smart Contract Framework

Finn Jarvi
*Stanford University*
Stanford, CA, USA

Cash Hilinski
*Cornell University*
Ithaca, NY, USA

***Abstract*—The health care claims process in the United States is fundamentally flawed, with approximately 450 million healthcare claims denied every year and appeals taking months. This paper presents a system designed to streamline this process. More specifically, this proposed system's goal is to leverage the decentralized nature of blockchain technology to create a parameterized smart contract system that can autonomously verify when contractual conditions are met and trigger instantaneously. To implement this, we used Ethereum for the blockchain, which is highly secure and supports smart contracts; the smart contracts are developed using neural network transformer models that autonomously template the contracts with data fetched from an EHR (Electronic Health Record) database, creating a custom smart contract relating to the policyholder's claim. When the customized predetermined conditions of the smart contract are met—for example, correct diagnosis and procedure codes—the smart contract autonomously triggers reimbursement. Initial testing has shown our proposed system is capable of processing claims end-to-end in under 60 seconds.**

***Index Terms*—blockchain, smart contracts, transformer models, electronic health records, healthcare claims, Ethereum**

## I. Motivation and Approach

The insurance industry in the United States utilizes a slow, inefficient, and grueling process to verify and pay out claims filed by policyholders [1]. When policyholders need to make a claim, they must first gather medical records, itemized medical bills, diagnosis and procedure codes, and other documents [2]. When insurance companies receive these documents, they must check the patient's identity, review policyholder status, verify the medical services are covered under the policy, certify the service was provided by an authorized provider, ensure costs fit within the reimbursement policy, verify copays, and check if there are applicable deductibles [3]. If there are any administrative errors, incorrect procedure or diagnosis codes, authorization errors, or other human errors, the claim will most likely be denied [4]. Attempting to appeal the claim typically requires the policyholder to restart the process [5]. This system is not only frustrating and inefficient but also a tragedy for Americans needing immediate care [6].

Our system proposed to address this problem shares similarities with the work of Chen et al. [7], who explored using blockchain's decentralized nature to eliminate fraud from insurance claims. While their approach addressed fraud in insurance claims using existing blockchain, it fails to address the 324 million claims that are initially denied because of clerical error and then later overturned. Chen et al.'s approach would also allow insurance companies to create and deploy smart contracts that have pre-determined conditions on the blockchain. When these conditions are met, the smart contract would trigger and reimburse the policyholder. While this process is effective at preventing fraud and making the reimbursement process more efficient, it does not actually solve the problem [8]; the smart contracts would still need to be reprogrammed by hand to fit complex claims, and policyholders would still need to gather extensive medical documentation to upload to an IPFS (InterPlanetary File System), which—depending on the complexity of the claim—can actually be counterproductive.

Our tested system takes Chen et al.'s innovative ideas a step further by removing off-chain storage for claim data and instead uses dynamic, real-time updating EHR databases. Unlike an IPFS, policyholders would not need to upload extensive documentation. We will develop a custom oracle that pulls real-time claim data from EHR databases. After the data is pulled, it will then feed into the prompts of coding-oriented transformer models to create the smart contracts using the EHR data specific to each policyholder's claim. The smart contracts created will have generated conditions based on the EHR data that, when verified, trigger the smart contract to reimburse the policyholder.

This solution eliminates the time-consuming process of collecting and verifying large amounts of medical documents and essential information [9]. Simultaneously, the decentralized nature of blockchains would help detect fraudulent healthcare claims [7]. Our system aims to streamline the entire process, allowing policyholders who desperately need reimbursement for medical treatment to not have to wait for their claim to be approved or risk their claim being denied due to human error.

## II. Project Logistics and Organization

The first step we took in implementing this solution was developing a custom oracle capable of retrieving data from multiple databases [2] based on the MRN (Medical Record Number) and claim-related information provided [1] and delivering it back to the Ethereum blockchain, where the smart contracts will be held. We built the oracle in Python because of its ability to seamlessly integrate APIs (Application Programming Interfaces) with its vast library ecosystem [10]. The oracle interacts with the EHR systems via the RESTful and GraphQL API endpoints. These endpoints allow for external systems, like our custom oracle, to query and retrieve specific medical data [11]. In our case, the policyholder will need to submit their MRN, and the oracle will find all the data related to the policyholder and their claim. When the data is in transit, it will be secured with AES-256 and authorized with OAuth 2.0. This helps ensure that the oracle meets standard HIPAA requirements [12].

When the oracle retrieves the claim-related data (medical records, diagnosis and procedure codes, and other important documents), it processes the information [3] and validates it. This validated data then gets fed into a code-oriented transformer model (Claude 3.5 Sonnet), which uses the claim-related information and creates a smart contract draft [4] with specific parameters (policyholder status, patient identity, authorized provider, and many other parameters). When the smart contract is generated, the insurance company can briefly review the parameters [5] and then deploy it to the Ethereum blockchain [6]. After deployment, the smart contract will once again verify that all the parameters are met and then trigger reimbursement [13] [7].

Initially, our system was only capable of connecting to a singular simulated EHR database via RESTful and GraphQL endpoints, and automating basic insurance claims. These were our milestones:

1) **The oracle is capable of connecting simultaneously to multiple databases.** First, we need to verify that each database we want to connect to has RESTful or GraphQL endpoints. Next, we need to develop "reusable" API connection modules, ideally each having a different class with separate methods of authentication and querying.
2) **The oracle is able to process, combine, and verify data retrieved.** Once the data is retrieved, it should be mapped to a "standard" format. All duplicate and irrelevant information to the claim should be thrown out, while all relevant information should be kept and formatted. This process will take place in the transformation pipeline. First, we will create the format (stored in a JSON schema), then create the "rules" (stored in another JSON file) to put data into that schema. Then the transformation pipeline will map that data to the schema in accordance with the "rules." Finally, the mapped data will be verified against the schema and be prepped to be transferred to our transformer model.
3) **The pipeline to feed data to the transformer model is created.** The mapped data will be fed to the transformer model (Claude 3.5 Sonnet) via API POST requests. This will only trigger if all required information ("rules") for the claim is filled and verified (this conditional check ensures only complete claims are submitted).
4) **The transformer model drafts the first smart contract and documentation is complete.** This will mark the end of our project. The documentation will need to be precise and provide in-depth explanations of regulations and backend components.

TABLE I
Comparison of Claims Processing Metrics

| Metric | Traditional[1] | Automated System |
|---|---|---|
| Error rate (%) | 19.3 | 1.2 |
| Cost per claim (USD) | 25.00 | 2–5 |

[1] Traditional data adapted from the American Medical Association.

We began by testing each individual component, such as the API connections, data mapping, and the transformation pipeline with placeholder data. When we confirmed that each individual component was operating correctly, we attempted testing combinations of different systems with placeholder data, such as API connection, data retrieval, and authentication. Then, we did a full end-to-end test with placeholder data. Next, we stress tested the end-to-end process with many claims being processed at once. Finally, we reviewed the tests to ensure all healthcare regulations were being met.

## III. Results/Conclusion

After extensive testing, the results are as follows: The end-to-end process, from the policyholder entering their MRN and basic claim information to the smart contract being drafted and ready to be briefly reviewed, has taken no more than 60 seconds. Additionally, we have seen the clerical error rate drop down to 1.2% over 2,000 synthetic and de-identified claims. Furthermore, taking into account gas and API fees, the cost per claim has decreased 85% from the traditional claims process. Stress testing has proved the system is capable of handling a substantial number of claims simultaneously.

While testing has shown the system to be reliable and effective at improving the insurance claims process, we have identified potential problems:

1) **Retrieving data becomes troublesome because of authentication and encryption requirements.** We plan to secure data during transit and at rest with OAuth 2.0, JWT, and AES-256 (Advanced Encryption

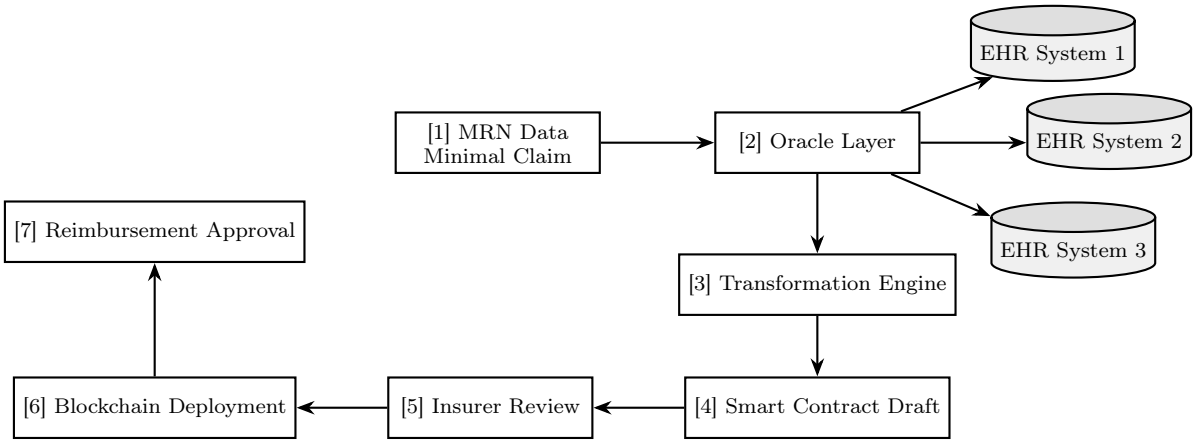


Fig. 1. System dynamics chart. In the paragraphs below, [#] refers to this chart.

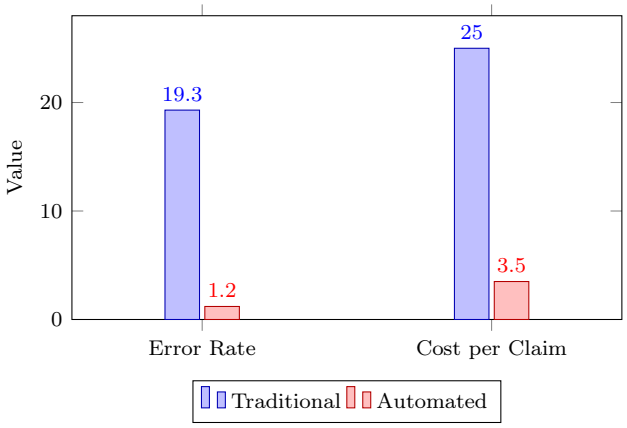


Fig. 2. Comparison of claims processing metrics. Traditional data adapted from the American Medical Association.

Standard, 256-bit version). OAuth 2.0 will allow us to access the EHR databases with ease while JWT ensures no unauthorized party is retrieving data [14]. In addition to this, we will use AES-256 to encrypt claim-related data the oracle retrieves. This is to ensure that all regulations are met.

2) **The oracle struggles to find claim-related information.** This has not been the case with our prototype. However, could this be an issue with actual EHR databases? We don't believe so. With the policyholder's MRN and the real-time updating of EHR databases, this is unlikely to be an issue. Nevertheless, if this becomes an issue, we can take in more claim-related information from the policyholder, such as a description of the treatment. This way, we can improve the oracle's ability to locate claim-related information without relying on extensive documentation input from the policyholder.

3) **Some EHR databases require APIs that our system does not support.** If the system needs access to a particular EHR database that does not support RESTful and GraphQL, we will need to construct reusable classes that can use other protocols. With these classes, anyone can easily edit them to support other communication methods.

The novel framework to automate insurance claims through EHR interoperability, transformer models, oracles, and smart contracts provided in this paper may be integrated into existing systems and built upon. The system has demonstrated significant reductions in processing time and clerical errors. As healthcare systems migrate to digital structures, there remains an urgent need for solutions that address administrative inefficiency. Future research may build upon this framework by incorporating additional interoperability standards and testing further in a clinical environment.

## Acknowledgment

We thank Paul Bottino, co-founder of the Technology and Entrepreneurship Center at Harvard (TECH), for his feedback and guidance on this paper.

## References

[1] R. Chandawarkar et al., "Revenue cycle management: The art and the science," *Plastic and Reconstructive Surgery Global Open*, vol. 12, no. 7, p. e5756, 2024, doi: 10.1097/gox.0000000000005756.

[2] U.S. Department of Labor, "Filing a claim for your health benefits," 2025. [Online]. Available: https://www.dol.gov/agencies/ebsa/about-ebsa/our-activities/resource-center/publications/filing-a-claim-for-your-health-benefits

[3] J. M. McCullough, "Timing of clinical billing reimbursement for a local health department," *Public Health Reports*, vol. 131, no. 2, pp. 283–289, 2016, doi: 10.1177/003335491613100212.

[4] L. Poland and S. Harihara, "Claims denials: A step-by-step approach to resolution," *Journal of AHIMA*, Apr. 25, 2022. [Online]. Available: https://journal.ahima.org/page/claims-denials-a-step-by-step-approach-to-resolution

[5] American Hospital Association, "Payer denial tactics — How to confront a $20 billion problem," May 6, 2024. [Online]. Available: https://www.aha.org/aha-center-health-innovation-market-scan/2024-04-02-payer-denial-tactics-how-confront-20-billion-problem

[6] W. E. Bennett, "Insurance denials of care amount to unlicensed medical practice," *Journal of Managed Care & Specialty Pharmacy*, vol. 26, no. 7, pp. 822–824, 2020, doi: 10.18553/jmcp.2020.26.7.822.

[7] C.-L. Chen et al., "A traceable online insurance claims system based on blockchain and smart contract technology," *Sustainability*, vol. 13, no. 16, p. 9386, 2021, doi: 10.3390/su13169386.

[8] S. Lee et al., "Privacy preservation in patient information exchange (PIE) systems based on blockchain: System design (preprint)," *Journal of Medical Internet Research*, vol. 24, no. 3, 2021, doi: 10.2196/29108.

[9] J. Hawayek and O. AbouElKhir, "Problems with medical claims that artificial intelligence (AI) and blockchain can fix," *Blockchain in Healthcare Today*, vol. 6, 2023, doi: 10.30953/bhty.v6.273.

[10] A. Reddy and Shilparani, "Open data and APIs—Data extraction and exploration using Python," in *Proc. 4th LIS Academy Conf. Open Scholarship and Libraries*, 2022. [Online]. Available: https://www.researchgate.net/publication/363510568

[11] J. Nan and L.-Q. Xu, "Designing interoperable health care services based on Fast Healthcare Interoperability Resources: Literature review," *JMIR Medical Informatics*, vol. 11, no. 1, p. e44842, 2023, doi: 10.2196/44842.

[12] J. Choi et al., "The OAuth 2.0 web authorization protocol for the Internet Addiction Bioinformatics (IABio) database," *Genomics & Informatics*, vol. 14, no. 1, pp. 20–27, 2016, doi: 10.5808/gi.2016.14.1.20.

[13] S. Lin et al., "SciviK: A versatile framework for specifying and verifying smart contracts," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.02209

[14] S. V. V. Machapatri and S. Gorantla, "Seamless integration of EHR systems with Salesforce Health Cloud: Revolutionizing patient data management," *Frontiers in Health Informatics*, vol. 13, no. 7, pp. 469–501, 2024.

Blockchain · Computer Science · Industry

# Automating Healthcare Claims with AI and Blockchain

A Transformer-Based Smart Contract Framework

Finn Jarvi *(Stanford University)*

Cash Hilinski *(Cornell University)*

With thanks to Paul Bottino, co-founder of the Technology and Entrepreneurship Center at Harvard, for his feedback and guidance.

2025

This work was completed while Finn Jarvi and Cash Hilinski were Paulson Fellows at the Technology and Entrepreneurship Center at Harvard.