Homebase: A Solution to Surprise Billing

A multi-party business process automation tool for the medical billing industry

Falls Technology

January 16, 2023

1 The Problem: Surpise Medical Bills

It is not uncommon for someone to undergo a surgery in May, pay all their bills by July, only to unexpectedly receive a bill for $3,000 in October. This is an example of surprise billing, a problem so pervasive that Congress passed the No Surprises Act in 2020, aimed to protect healthcare consumers from such bills.[1] In 2015, "Consumer Reports found 30 percent of Americans with private health insurance have received surprise bills, where their insurance plan paid less than they expected. Of those, 23 percent received a bill from a doctor they didn't expect to get a bill from. And 14 percent said they were charged higher out-of-network rates by doctors they thought were in-network."[2]

Kaiser Health News has an entire series dedicated to surprise medical bills titled "Bill of the Month." See *https://khn.org/*.

For many people, surprise medical bills can destroy them financially. Consider the experience of Michael: "I went to a 30-day rehab a while back, it was covered completely. When I finished the program they asked me to attend outpatient which again they told me was covered, not even a copay. Years later I have a $40,000 medical bill on my credit report and I have no intention of paying it and no idea how to deal with it. So much for buying a house."[3]

Scenarios like Michael's occur too frequently in a world that has highly sophisticated software; there should be no reason why Michael can't be updated with each transaction between himself, the hospital, and his insurer. However, healthcare IT is plagued with siloed systems that fail to interoperate without significant friction. Such friction is not surprising when there is a lack of trust between all parties involved in a particular transaction.[1] While transparent medical billing will provide reassurance to the patient, it will also create more trust between medical providers and payers–the insurance companies. Insurers have to often adjudicate health insurance claims to ensure that the provider has not "updcoded" or charged for multiple services.

[1] There are many ways that providers, patients, and insurers can commit health care fraud. Providers can submit multiple bills for the same service, and patients fraudulently use someone else's insurance. See *https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-and-safety/common-scams-and-crimes/health-care-fraud*.

Consider another experience, that of Leanne from Chicago: "We were pregnant with our first baby last year, and we called

[1] See https://www.congress.gov/bill/116th-congress/house-bill/3630/text.
[2] See https://www.pbs.org/newshour/health/americans-who-confronted-surprise-medical-bills-share-their-stories.
[3] *Ibid.*

our insurance co several times to confirm how much our out of pocket max would be. They said $4,000 max, for mom and baby together, assuming normal birth and normal hospital stay of 48 hrs. All of it should be billed to my insurance deducible. We asked again and again to be sure. No other charges should occur, they reassured us. Always got the same answer. So we saved and saved to make $4,000 happen. After the birth, the hospital billed my newborn's nursery stay charges during the birth days to HIS insurance, not mine, causing another $1,000 or so in costs to us. I was told by several insurance customer service people that this was an error. We filed several appeals which were denied. I was livid. It cost us almost a month of my maternity leave that we are funding ourselves. It makes no sense but our hands are tied since the hospital billed it and insurance processed it."[4]

If Leanne received real-time updates on the transactions and queries made on her behalf, she could quickly notify the relevant parties of the crucial mistake. Moreover, with zero-knowledge proof technology, a specific transaction could be prohibited if the identity of the policy-holder does not match the patient involved. While the No Surprises Act intends to protect people from experiences like Leanne's, it will be difficult to enforce and monitor without a system in place. It is our hope that Homebase becomes such a system.

As a multi-party business process automation (BPA) tool, Homebase relies on zero-knowledge proof technology and a pub-sub, notification-based middleware to verify contract compliance between parties and provide real-time updates of transactions to all subscribing parties. With Homebase, Leanne would be notified each time her data was transacted on, relaying the items of her medical bills and insurance claims at a level as granular as the parties agree upon.[2] In the following sections, we detail how Homebase works and prove why it can solve the issue of surprise billing.

[2] For each transaction, Leanne could see data passed between the Hospital and Insurer. Homebase relies on RISC Zero's zero-knowledge virtual machine for producing zero-knowledge proofs as well as *journals*, which Leanne would see on her records. These details are expanded upon below.

2 The Solution: Homebase

We named the system Homebase to symbolize a particular party's data rounding all bases securely and efficiently, undergoing the requisite computations in order to come home, visible in their own system of record. The Homebase system comprises a RISC Zero zkVM, a Data Processor, and a UI. The

[4] Ibid.

Data Processor will implement OData.Neo, a new, protocol-agnostic version of OData that can ideally query systems that may use non-REST based API's–such as gRPC or SOAP.[3] The pub-sub, notification-based middleware Homebase uses is based on OData.Neo. The Data Processor will handle bi-directional activity that may either come from the external system or the UI. For instance, a patient may enter data into the Homebase UI–which triggers synchronization with their medical provider's database, requiring a sequence of verifiable computations processed by the zkVM. Or, an insurance provider may enter a claim within their external system which automatically becomes processed by the Data Processor, which passes the relevant data to the insurance provider's zkVM.[4] Please see Figure 1 below to reference the Homebase system architecture.

[3] OData is the Open Data Protocol, initiated by Microsoft in 2007, standardizes how disparate RESTful API's are consumed and integrated with eachother.

[4] The specific operations of Homebase will be explained further in the following sections.

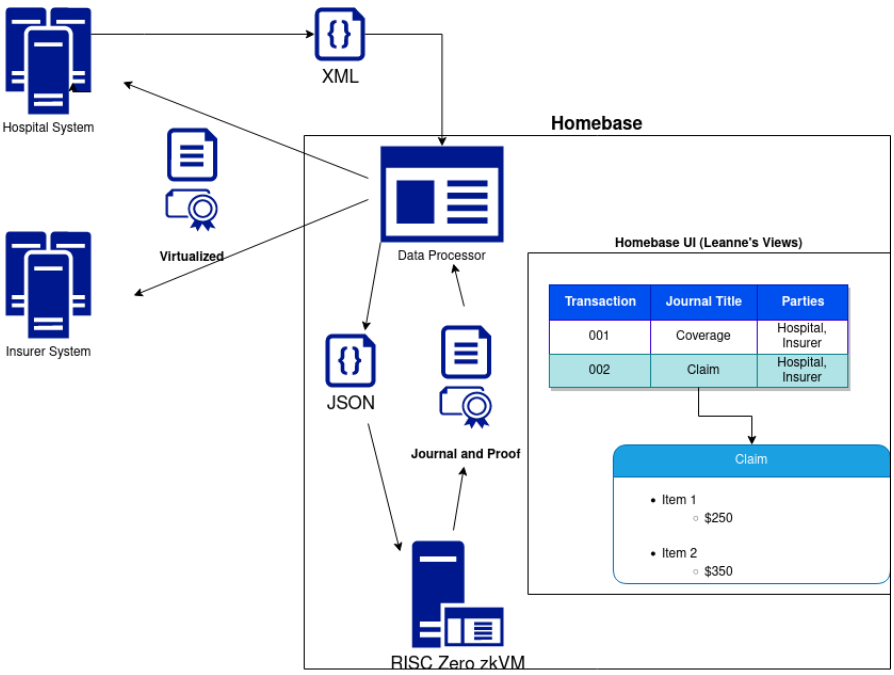

Figure 1: Homebase architecture. The hospital sends a POST request with Leanne's health insurance claim as XML body. The Data Processor then converts the data into JSON format that can be given to the zkVM. The zkVM returns a zero-knowledge proof and journal that will then be passed back through the Data Processor for distribution to all parties.

2.1 RISC Zero and the Zero-Knowledge Virtual Machine (zkVM)

The RISC Zero zkVM provides verifiable computation through an implementation of zero-knowledge proofs (zkp's) via the

RISC-V instruction set architecture (ISA).[5] RISC Zero's zkVM receives a guest method from a particular party's code-base and produces a proof of the result by first, converting the guest method into an executable file to be executed according to RISC-V assembly code, then, translating the guest method into polynomial form so that it is digestible by the zkp protocol.[5]

[5] A zkp is a cryptographic protocol that proves the truth of a particular claim or computation without revealing any information. For instance, with a zkp, a prover can prove that a particular value is present within a database without revealing any information about the value or the database itself.

2.1.1 Proving Compliance with the zkVM

While the healthcare sector has been attempting to enhance interoperability through HL7 FHIR (henceforth FHIR), international standards of health data formatting and transmission, technologies that truly take advantage of such standards are few and far between. Homebase intends to take full advantage of the standard developed by FHIR by supplying proofs of compliance and transparent journals as a result of combining standardized methods with the power of the zkVM.

HL7 FHIR (Health Level 7, Fast Healthcare Interoperability Resources) has developed a data model that acts as the standard for formatting and transmitting medical data. Such data could be regarding workflow, financial, or legal data specifically. Please see *http://hl7.org/fhir/*.

For example, FHIR specifies stateful *Contract* and *Coverage* resources, or objects, that can exist for multiple parties. Represented as a JSON object, a contract $Contract_{Hospital,Insurer}$ and coverage $Coverage_{Leanne,Insurer}$, can be referenced when proving compliance of a specific health insurance claim $Claim_{Leanne}$. To be concrete, if Leanne's hospital were to file the FHIR-compliant claim in 1, the zkVM could prove:

1. Leanne is indeed the correct patient in question.
2. Code 62264, "Surgical Procedures on the Spine and Spinal Cord" is covered under $Coverage_{Leanne,Insurer}$
3. $12,500 is the correct amount according to $Coverage_{Leanne,Insurer}$ and $Contract_{Hospital,Insurer}$
4. If necessary, $Contract_{Hospital,Insurer}$ allows for coverage of 62264-coded services.

The reason we can prove the above claims is that the guest code, to be run by the zkVM, will call all resources, utilize its

[5] The technical aspects of how exactly code can be translated into algebraic forms is left out of this paper. But there are plenty of great resources for the reader to reference, including the following:

- *https://medium.com/starkware/stark-math-the-journey-begins-51bd2b063c71*
- *https://www.risczero.com/docs/explainers/proof-system/proof-system-sequence-diagram*

business logic to traverse each JSON Object, and output in the journal whatever data the parties agree to reveal.[6] In general, for any key-value pair within a tree-like structure–whether that be a nested JSON or XML tree–the zkVM can verify any program intended to compare or validate one key-value pair against another. This comparison and validation is often what comprises most multi-party business processes. Homebase hopes to automate and make such processes transparent.

[6] For another concrete example that is also fun and comes with actual code, please see RISC Zero's implementation of battleship here: *https://github.com/risc0/battleship-example*.

Listing 1: Leanne's Insurance Claim

```
{
    "resourceType" : "Claim",
    "id" : "9876B1",
    ...
    "contained": [
    {
      "resourceType": "Patient",
      "id": "patient-1",
      "name": [
        {
          "use": "official",
          "family": "Ashcraft",
          "given": [
            "Leanne"
          ]
        }
      ],
      "gender": "female",
      "birthDate": "1954-06-11",
      ...
    "productOrService": {
        "coding": [
          {
            "system": "http://www.ama-assn.org/go/cpt",
            "code": "62264",
            "display": "Surgical Procedures on the Spine and Spinal
                Cord"
          }
        ]
      },
      ...
  "total": {
    "value": 12500.00,
    "currency": "USD"
      }]
    }
```

3 The Data Processor and OData.Neo

While the zkVM is vital to Homebase delivering verifiable computation, it would be completely useless if Homebase couldn't then share messages and receipts to sync it's connected parties' databases and systems of record. As mentioned above, we have determined that OData.Neo is the best protocol and

open source project for managing the sharing of data, including the proofs and journals produced by the zkVM.[7]

Based on OData, OData.Neo is protocol agnostic, meaning that it can interoperate with systems that may use RESTful, gRPC, SOAP, or GraphQL. This is important because an insurance company can interoperate with a hospital system that uses SOAP, and a physical therapy office that uses a simple RESTful API. To implement protocol-agnostic operations, OData.Neo converts OData queries into OData.Neo expressions, a common format that can then be translated to any expression or query designed to interact with disparate DBMS's. Our Data Processor highly relies on OData.Neo to provide most of the logic for Homebase's pub-sub, notification system.[6]

Specifically, OData.Neo's *Substrate* component acts as an event bus that components subscribe to. So, a hospital system can send a claim and a subscription can be passed back to the system which notifies the system of changes that are made to a specific Query being processed within OData.Neo. Refer to Figure 2 below to see how the Substrate interacts with the rest of the Data Processor.[8]

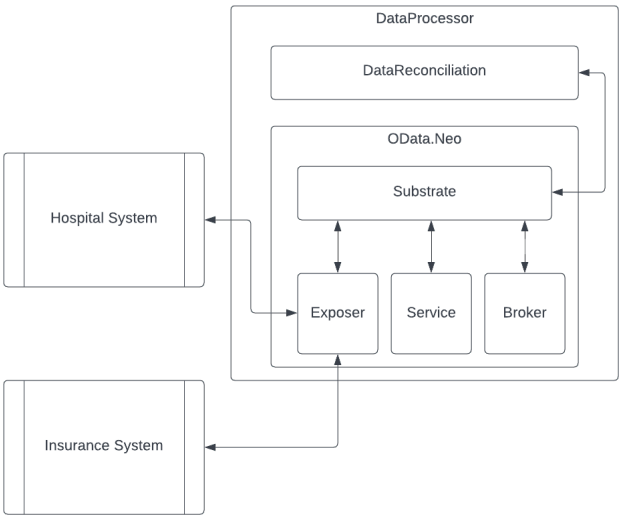

Figure 2: Data Processor architecture.

[7] OData.Neo is also built according to *The Standard*, a software design standard developed by Microsoft's Hassan Habib and company. Three components are required for Standard-compliant projects: Brokers, Services, and Exposers. All three are vital to OData.Neo and Homebase.

[8] Most importantly, a patient like Leanne can subscribe to the Substrate and be notified about any changes made to the query involving her data.

[6] OData.Neo is still under construction and we are constructing OData.Neo.Java–the Java implementation of OData.Neo. OData.Neo's architecture is deeper and more complex than what is described here, but we hope that the reader has gained some understanding of how OData.Neo will be a vital component to the operation of Homebase.

Service, Broker, Exposer, and Data Reconcilation Components The Service component, which interacts with the Substrate, is responsible for executing the core business logic of OData.Neo. This is where the primary contracts are formed for generic OData.Neo entities and OData expressions are formed. Ideally, and according to Standard requirements, the Service component will be isolated from external dependencies as much as possible. OData.Neo's Service component is complicated and a full explanation would require another five pages.

If a particular process requires an external library or particular API, the Broker component comes in to help. Assisting in the isolation of the Service component from external dependencies, the Broker responds to events in the Substrate and passes functionality into the Substrate to be used in the Service's computations. For instance, assume that the Substrate passed a hospital's insurance claim query to the Service component, which has begun mapping the query into an OData expression. However, this insurance claim requires adherence to HL7 FHIR formatting. In order to ensure compliance to HL7 FHIR, the Broker makes a call out to *http://terminology.hl7.org/CodeSystem/v3-ActCode*, for instance, to confirm the Service-constructed expression does so.

Once the OData.Neo expression is constructed, it is passed to the Data Processor's DataReconciliation component to interface with the rest of the Homebase application, especially the zkVM functionality. OData.Neo's Exposer component is responsible for determining the protocol used for different parties' systems, so that the DataReconciliation component can execute the necessary operations for Homebase to interact with the external systems.

4 Related Work: Similarities and Contributions

Much of the research and work surrounding multi-party workflow automation involves an intersection of blockchain and business process automation (BPA).[7] The existing applications and research around the technology seem to not rely on

[7] See Flavio Corradini et al's *Model-driven engineering for multi-party business processes on multiple blockchains*, and Simm, Steiner, and Truu's *Verifiable Multi-Party Business Process Automation* for example.

zero-knowledge proof technology, except for another OASIS standard called the *Baseline Protocol*.[9] However, Baseline's use of zero knowledge proof technology is not as mature as RISC Zero and does not utilize a zkVM within its architecture.

There are however some examples of utilizing zero knowledge proof technology for healthcare applications.[8] RISC Zero's zkVM does not assume any network-specific protocol, so we are not tied to blockchain technology like most related applications.

[9] The Baseline protocol does not specify the use of any particular blockchain network or zero-knowledge proof technology. It is intended to purely be a standard.

5 Conclusion

The technology that Homebase is building upon is very new and exciting. Everything Homebase is involved in is under active development. We hope that with Homebase, patients can be free from debilitating surprise medical bills, staying updated on any computation involving their medical data.

Moreover, on a larger scale, our hope is to bring a technology to businesses and individuals that makes working with one-another easier and reduces friction. Our long term goal is to help optimize all healthcare services, as well as make intergovernmental processes less cumbersome. In short, we want to slice through areas of intractable bureaucratic gridlock and provide the world with more opportunities to innovate and solve harder problems rather than stay stuck working on repetitive workflows.

[8] Please see Houyu Zheng et al's *A novel insurance claim blockchain scheme based on zero-knowledge proof technology*, and Medibloc, https://medibloc.com/en/, a blockchain-based health data exchange.