openIDL ND Uninsured Motorist POC

Architecture, Security and Data Privacy

Table of Contents

Document Control

Date	Description	Author
2022-08-29	Initial	Ken Sayers

Overview

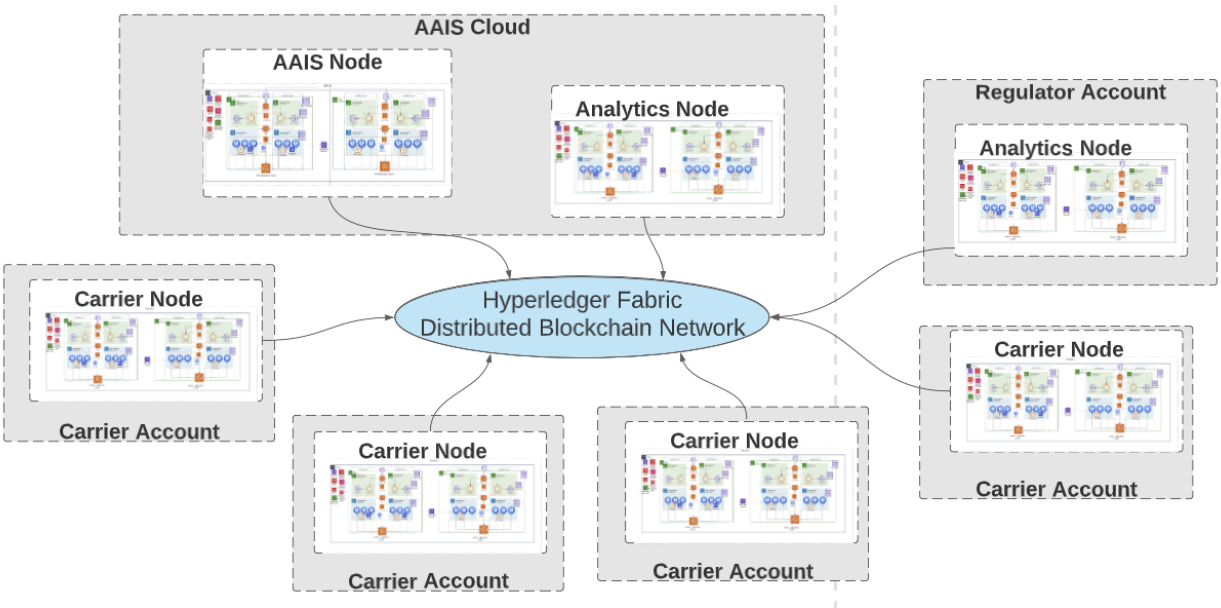
openIDL is an information exchange platform purposefully designed and built to improve the regulatory reporting process for the insurance industry. openIDL is a network of nodes that communicate through Hyperledger Fabric to increase data integrity, security and privacy for regulatory reporting in the insurance industry. Here we provide some details how this data integrity and privacy are supported by illustrating the architecture and pointing out its key features.

Architecture

Network

The openIDL system is a network of nodes. There are three kinds of nodes:

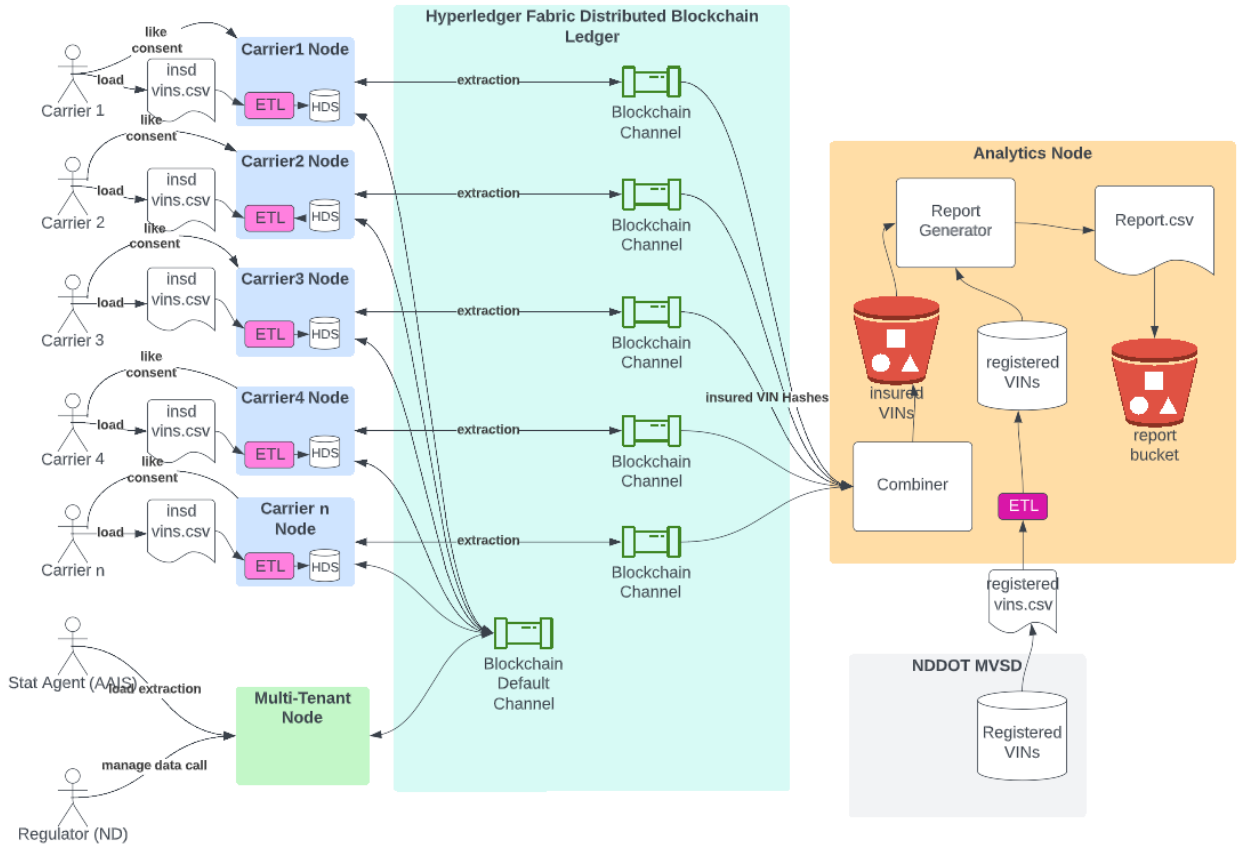
- **AAIS Node** - a multi-tenant node which can be used by multiple carriers, the DOI and stat agents.
- **Analytics Node** - The recipient of data extractions and where the report creation happens.
- **Carrier Node** - a single-tenant node that holds the carrier harmonized data in a private account owned by the carrier.



Each grey box is hosted in a different account (in AWS this is an organization). See more about AWS organizations here: https://aws.amazon.com/organizations/

POC Network

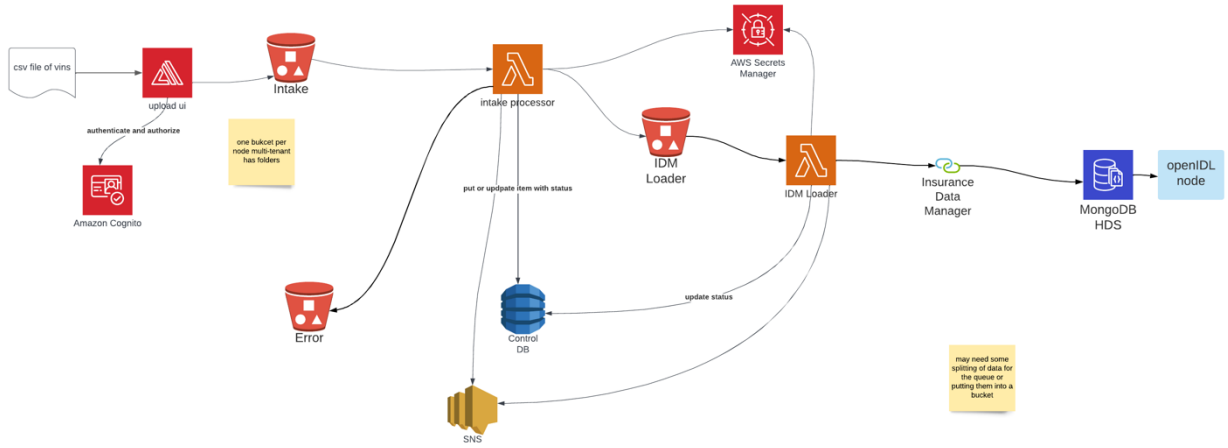
The specific network for the ND POC looks like the following:



There are ***n*** carrier nodes. One for each carrier. The insured VINs are loaded into the node by the carrier. During the load, the VINs are hashed using a salted SHA256 and the hashes are stored alongside the VINs in the HDS. Hyperledger Fabric channels are used to exchange data between nodes. The business of communication about the data call happens on the default channel. The data, when extracted, is passed to the analytics node over a peer-to-peer channel between the carrier and the analytics node so information exchange from a carrier to the reporting fuction is never visible to any other carrier or node on the network. The analytics node combines the extracted data from all the carriers, compares the extracted insured VIN hashes with the registered VIN hashes and produces a report of matches. VIN numbers are never transferred or analyzed as plain text, only as encrypted values.

ETL

The flow of data into the HDS follows this architecture:



An application is accessed via a URL. This is built with Amplify and uses Cognito for authentication. This is the same Cognito service used by the data call application. The file is placed into the intake bucket. The intake processor is triggered when files arrive in the intake bucket. The intake processor checks for errors and sends errored records to the error bucket. The control database is updated with the processing status of the file and the notifications are sent using the Amazon Simple Notification Service (SNS). Valid records are loaded into the IDM Loader bucket where they are picked up by the IDM Loader. This lambda sends the records into openIDL through the Insurance Data Manager API which loads them into the Harmonized Data Store.

Foundational Technologies of the Architecture

openIDL is built on a solid foundation. There are five main elements:

- Amazon Web Services
- Open-Source Technologies
- Kubernetes
- Hyperledger Fabric
- Infrastructure as Code

Amazon Web Services

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 200 AWS services are available. New services can be provisioned quickly, without the upfront fixed expense. This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements.

Here are some of the services used for an openIDL node:

Service	Description	openIDL Use
Amplify	AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve. With Amplify, you can configure a web or mobile app backend, connect your app in minutes, visually build a web frontend UI, and easily manage app content outside the AWS console.	Upload UI
API Gateway	Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications. API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authorization and access control, throttling, monitoring, and API version management.	Upload UI

Service	Description	openIDL Use
Certificate Manager (ACM)	AWS Certificate Manager is a service that lets you easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the Internet as well as resources on private networks. With AWS Certificate Manager, you can quickly request a certificate, deploy it on ACM-integrated AWS resources, such as Elastic Load Balancing, Amazon CloudFront distributions, and APIs on API Gateway, and let AWS Certificate Manager handle certificate renewals. It also enables you to create private certificates for your internal resources and manage the certificate lifecycle centrally.	Domain certificates
CloudTrail	AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of an AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command-line tools, and other AWS services	It captures all the resource audit logs as part of the deployments for audit purposes.

Service	Description	openIDL Use
CloudWatch	Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.	It collects standard metrics and logs for AWS services
Cognito	Amazon Cognito provides user sign-up, sign-in, and access control to web and mobile apps quickly and easily. With Amazon Cognito, you also have the option to authenticate users through social identity providers such as Facebook, Twitter, or Amazon, with SAML identity solutions, or by using your own identity system. In addition, Amazon Cognito enables you to save data locally on users' devices, allowing your applications to work even when the devices are offline. You can then synchronize data across users' devices so that their app experience remains consistent regardless of the device they use.	Identity provider for openIDL APIs and User Interfaces.
DynamoDB	Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-master database with built-in security, backup and restores, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and support peaks of more than 20 million requests per second.	Manage terraform state file locking Control DB for ETL

Elastic Cloud Compute (EC2)	Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. The simple web interface of Amazon EC2 allows you to obtain and configure capacity with minimal friction, providing complete control of your computing resources. **EC2 Auto Scaling** Amazon EC2 Auto Scaling helps you maintain application availability and allows you to automatically add or remove EC2 instances according to conditions you define. **Elastic Load Balancing** Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make an applications fault tolerant. Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request. Network Load Balancer (NLB) is best suited for load balancing of TCP traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.	Instances • Bastion hosts • EKS worker nodes NLB • Application load balancer • Network load balancer EBS • root volumes

Service	Description	openIDL Use
	Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network. **Elastic Block Store (EBS)** Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect from component failure, offering high availability and durability.	

Service	Description	openIDL Use
Elastic Kubernetes Service (EKS)	Amazon Elastic Kubernetes Service (Amazon EKS) makes it easy to deploy, manage, and scale containerized applications using Kubernetes on AWS. Amazon EKS runs the Kubernetes management infrastructure across multiple AWS availability zones to eliminate a single point of failure. Amazon EKS is certified Kubernetes conformant. Applications running on any standard Kubernetes environment are fully compatible and can be easily migrated to Amazon EKS.	All OpenIDL Application components deployed on EKS
Identity and Access Management (IAM)	AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.	Control access to AWS services used in the openIDL node.
Key Management Service (KMS)	AWS Key Management Service (KMS) makes it easy for you to create and manage keys and control the use of encryption across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses FIPS 140-2 validated hardware security modules to protect your keys. AWS KMS is integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs.	Manages all the data encryption keys
Lambda	AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you use.	Upload UI and ETL

Service	Description	openIDL Use
Organizations	AWS Organizations helps you centrally manage and govern your environment as you grow and scale your AWS resources. Using AWS Organizations, you can programmatically create new AWS accounts and allocate resources, group accounts to organize your workflows, apply policies to accounts or groups for governance, and simplify billing by using a single payment method for all of your accounts.	Manage carrier nodes
Route 53	Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. Amazon Route 53 effectively connects user requests to infrastructure running in AWS—such as EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets—and can also be used to route users to infrastructure outside of AWS.	Provide access to the externally available endpoints for the system. These include the insurance data manager and the openIDL UI.
Simple Storage Service (S3)	Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.	Manage terraform state files Manage terraform Input file
Secrets Manager	AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text.	Hold onto secrets during IaC operations.

Service	Description	openIDL Use
Simple Notification Service (SNS)	Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.	ETL file load notifications.
Virtual Private Cloud (VPC)	Amazon Virtual Private Cloud (Amazon VPC) enables you to define a virtual network in your own logically isolated area within the AWS cloud, known as a *virtual private cloud (VPC)*. You can launch Amazon EC2 resources, such as instances, into the subnets of your VPC. *Transit Gateway* AWS Transit Gateway is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a single gateway.	Provides an isolated area for openIDL nodes within the AWS cloud. It enables complete control over your virtual networking environment including resource placement, connectivity, and security. openIDL node has 2 VPCs spans across 2 available zones. • Application VPC • Blockchain VPC Application VPC: where we deploy all the non-blockchain deployments. BlockchainVPC: It will have all the HLF components and Hashi corp Vault cluster.
NON-AWS Technologies		

Service	Description	openIDL Use
Ansible	Infrastructure as Code	Lower level provisioning and configuration that Terraform is not suited for.
GitHub	Online git provider	Host opensource code for openIDL applications and infrastructure as code. Build and deploy application code. Provision and update network and nodes.
Jenkins	The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.	IaC
Hashicorp Terraform	Terraform is an open-source infrastructure as a code software tool that provides a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.	Provisioning of cloud resources.
Hashicorp Vault	Manages key-value pairs for secrets.	A 3 node vault cluster is deployed to manage secrets used in Infrastructure as Code and applications for authentication and permissions as well as other configuration tasks.

Service	Description	openIDL Use
Helm	Helm is an open-source packaging tool that helps you install and manage the lifecycle of Kubernetes applications. Similar to Linux package managers like APT and Yum, Helm manages Kubernetes charts, which are packages of pre-configured Kubernetes resources.	Configure Kubernetes and deploys OpenIDL application components.
FluxCD	Flux is a tool for keeping Kubernetes clusters in sync with sources of configuration (like Git repositories), and automating updates to configuration when there is new code to deploy.	IaC
mongoDb	A noSQL db. Not an AWS service. Provisioned on an EC2 instance directly.	Harmonized Data Store
Shell Scripts	Execute Linux commands for various reasons.	Configure hyper ledger fabric network and other utility scripts.

Open-Source Technologies

openIDL uses a number of open-source technologies to make the system transparent and easy to implement in different contexts. The main open-source used are:

- MongoDB to hold harmonized data
- Node JS for the component software language
- Angular for the User Interface
- Express for API
- GitHub for Distributed Version Control
- Infrastructure as Code - Jenkins, Ansible, Terraform and Vault

The Linux Foundation manages openIDL. All code for the applications and infrastructure-as-code is available in GitHub as open source.

Kubernetes

openIDL uses Kubernetes as the main container workload management. Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Some specifics of how Kubernetes is used for openIDL:

- There are two clusters used by openIDL
- One holds the HDS and the application code
- The other holds the Hyperledger Fabric components
- Jobs, services, pods, ingress and egress are all used to good effect
- The IaC holds all the details of how Kubernetes resources are configured

Hyperledger Fabric

Hyperledger Fabric is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability. It is designed to support pluggable implementations of different components and accommodate the complexity and intricacies that exist across the economic ecosystem.

How openIDL uses Hyperledger Fabric:

- **Private, Permissioned network** - all organizations are known to the network and are provided certificates by the certificate authorities. All communications on the network require the certificate. No organization can access the ledger without a certificate and permission from the network certificate authority.
- **Distributed Ledger** - The ledger (the blockchain) is distributed to each of the organizations. The copies of the ledger are managed by Hyperledger Fabric automatically.
- **Smart Contracts** - Hyperledger Fabric implements smart contracts using chaincode. openIDL chaincode is written in Go.

- **Channels** - Hyperledger Fabric supports the concept of channels. Channels are private subnetworks that allow individual organizations to participate with other organizations in private conversations. Each channel has its own ledger and private data collection. The default channel includes all organizations. Each carrier node participates in a peer-to-peer channel with the analytics node.
- **Private Data Collections (PDC)** - A private data collection is a Hyperledger Fabric feature that allows off chain data to be shared securely with other organizations in the network. The PDC is private to each channel. openIDL uses PDCs to share data from carrier nodes to the analytics node. The shared data is never more than the result of an extraction pattern. **Extraction Patterns** are not a Hyperledger Fabric feature. These are map reduce functions that are stored on the ledger with the data call to ensure that the query run against the HDS is immutable.

Infrastructure as Code

Infrastructure as Code is a way to capture all the setup of the infrastructure in configuration files and scripts. This "Code" is then managed in a version control management system like GitHub alongside the application code. This allows the setup to be transparent, robust and repeatable. openIDL uses Jenkins, GitHub, Terraform, Ansible and Helm for IaC. All the IaC is managed in GitHub.

Data Privacy

Here are a few details on how openIDL supports data privacy.

- The Harmonized Data Store is owned by the carrier
- It resides on the carrier node in the carrier's hosted account
- The data can be loaded through ETL using the same user ids for the data call application
- The database is not visible by anyone but administrators of the account
- No carrier can see any other carriers' data, they are in separate AWS accounts
- Chainyard, our trusted partner and yours in administration of the nodes is the only entity that has administration access to the accounts
- Administration access can be granted to the carrier on request
- When an extraction occurs, on consent by the carrier, the HDS is accessed, and the results of a map/reduce function (not the raw data) are sent to the analytics node over a private Hyperledger Fabric channel. We call this the "**Extraction Pattern**".
- The analytics node combines results from all extractions (consented by carriers) to generate the report. Once completed, the individual results are deleted from the analytics node and the private channel. A hash of the data is held on the ledger to keep evidence of the event.

Security

Here we provide some notes on how openIDL supports security:

- Each carrier has their own hosted AWS account.
- IAM credentials are provided only on request.
- Chainyard has credentials for these accounts.

- AAIS may have credentials if the carrier agrees
- Access to the database is only through the Insurance Data Manager API which is permissioned using Amazon Cognito
- Cognito IDs can be used to login to the upload UI which eventually feeds the HDS.
- The same Cognito IDs are used to view, like and consent to Data Calls.

Infrastructure Security

For the POC, using AWS organizations allows us to manage each node like a separate cloud. IAM roles and users are set up for the organization and control access to administration of all infrastructure in the node.

Expertise

AAIS has partnered with Chainyard to provide expertise in the technical foundations of the system. Chainyard also has expertise in penetration testing. The included security report is a product of Chainyard's security testing process.

SAST and DAST Testing

Please reference the included security report for information about the security testing of openIDL.

Conclusion

openIDL was purposefully designed and built to improve the regulatory reporting process for the insurance industry. The network has been built using Hyperledger Fabric to ensure data privacy, security and integrity. The system uses open-source technologies, and all code is open-sourced to provide transparency. Kubernetes is utilized to provide a robust container management layer, for easy installation and management. Infrastructure as Code is used to provide further transparency of the technical configuration. All nodes are implemented in AWS using best practices for security to provide a robust cloud-based solution that is secure.

Resources

Use these resources to find more information and the open source code.

- openIDL.org - https://openidl.org/
- Wiki - https://wiki.openidl.org/
- github
  - Main code base - https://github.com/openidl-org/openidl-main
  - Infrastructure as Code - https://github.com/openidl-org/openidl-aais-gitops
  - ETL - https://github.com/openidl-org/openidl-etl
- Documentation - https://openidl-documentation.readthedocs.io/en/latest/index.html