Traceability Driver Service

The Traceability Driver is a service that can be installed into an existing traceability solution that allows for the solution to become interoperable with other traceability solutions that also implement the Traceability Driver.



# Vocabulary

Below is a list of terms used throughout this document.

**Traceability Data** – This is a combination of master data and event data.

**Solution Provider** – This refers to a traceability software system that implements the Traceability Driver.

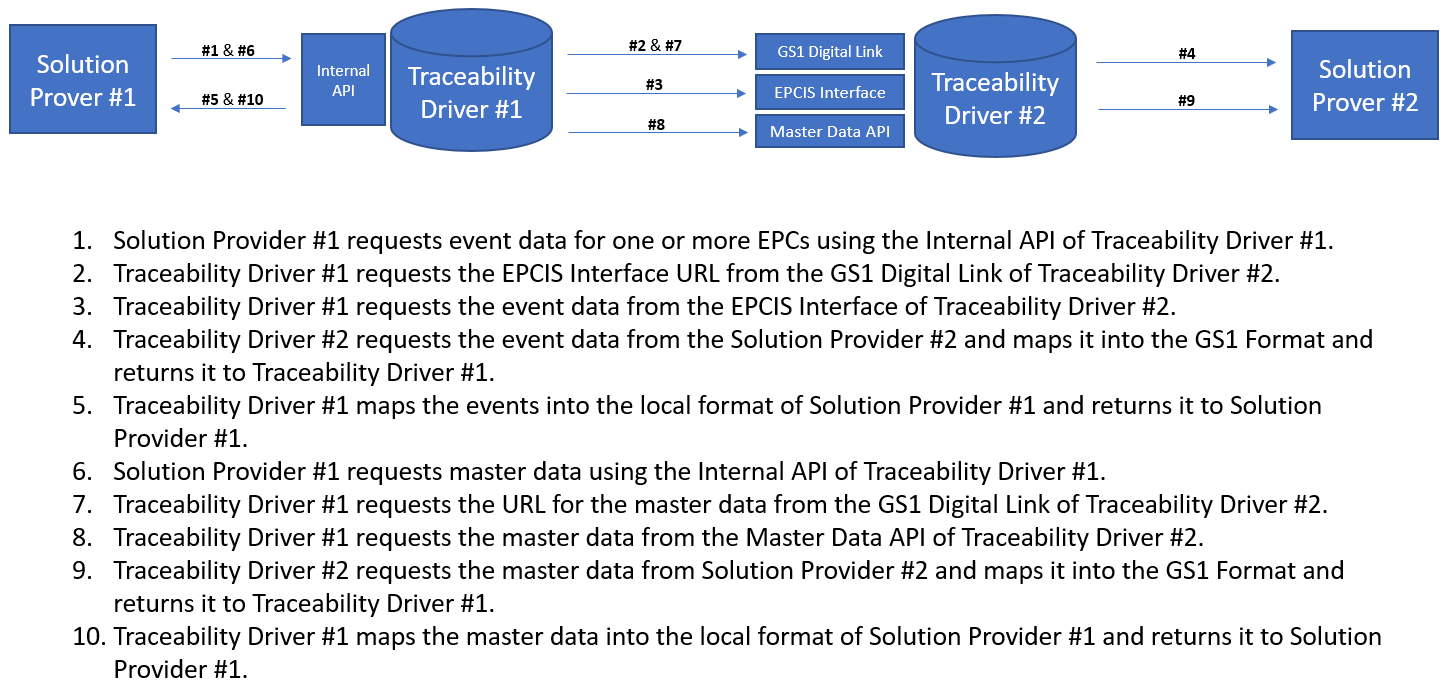
**Account** – This is a business entity / company that is hosted by a Solution Provider.

**Trading Partner** – This is a business entity / company that exchanges traceability data with an account.

# What does the Traceability Driver Offer?

* Greatly Reduces the cost of Interoperability
  + *Solution providers only need to write a single mapper from their internal data format to the common data models.*
  + *Solution providers only need to integrate with the simple REST APIs provided by the traceability driver.*
* Standardizes Static Data
  + *Fishing Areas*
  + *Fishing Methods*
  + *Harvest Methods*
  + *Etc..*
* Directory Service
  + *It is easy for accounts managed by the solution provider to look up and add accounts on other solution providers.*

# How does it work?



# Directory Service

The directory service is an optional service hosted by the Traceability Internet Organization. This service allows service providers to register accounts and search for other accounts that are managed by other service providers.

## Web Application

A C# Blazor Web Application is used to manage the directory service.

* Service Providers
  + *View Service Providers*
  + *Add new Service Provider*
  + *Change the Name of a Service Provider*
  + *Download the DID of the Service Provider*
  + *Regenerate a new DID for a Service Provider*
* Accounts
  + *View Accounts and their Associated Service Provider*

## Registering a Service Provider

Service providers will register by emailing the Traceability Internet Organization and requesting registration. A meeting will be scheduled in which the Service Provider would be required to demo their software, and this would provide a way to filter out bad actors from the Directory Service.

# Installation

The Traceability Driver is installed on any Mac / Linux / Windows computer as a web service. It is written in .NET 5 so that it is compatible on all three major operating systems. The traceability driver is unzipped into a folder on the target machine and the executable is ran providing an SSL Certificate and host URL.

## Configuration

The service is configured using a JSON configuration file that is provided upon start up of the service. A file watcher will be placed on this configuration file so that any changes made to the configuration file are immediately detected and uploaded back into the system.

* URL
  + *Configured on start up.*
  + *HTTPS / TLS is required.*
* SSL Certificate
  + *Configured on start up.*
  + *Required for HTTPS.*
* Solution Provider Public / Private Key (optional)
  + *Configured on startup but can be changed inside the application.*
  + *This is required if the service provider is looking to utilize the Directory Service.*
* Connection String
  + *Configured in the appSettings.json*
* Digital Links
  + *Configured in the appSettings.json under the “DigitalLinks” section.*
* Master Data Service URL Templates
  + *Configured in the appSettings.json.*
* EPCIS Query URL Templates
  + *Configured in the appSettings.json*
* Accounts
  + *Can be configured via the API provided.*
  + *Accounts can be viewed in the application provided.*
* Trading Partners
  + *Can be configured via the API provided.*
  + *Trading Partners can be viewed in the application provided.*
* Traceability Mapper / Driver
  + *The Assembly Name and the Class Name is configured in the appSettings.json*
  + *The DLL is placed in the “drivers” folder of the installation directory of the traceability driver.*

## Configuration Service

A configuration service will provide support for accessing the JSON configuration file and is injected into each controller using Dependency Injection. It will be added as a singleton service in the startup.

# Status Page

This is page that can be navigated to that shows the status of the Traceability Driver Service. This status page will show the following:

* The solution provider is registered and the provided public / private key is valid.
* The mapper was detected and is working.
* View logs.

# Accounts

An account will be generated by the solution provider for each business entity that they manage. An account consists of the following data points:

* ID
  + *This can be assigned by the solution provider or can be generated by the traceability driver internally. This ID must be unique across all accounts managed by the traceability driver instance.*
* PGLN
  + *This is assigned by the solution provider.*
* DID
  + *This is generated by the solution provider if the traceability driver is using the directory service.*
* Name
  + *This is assigned by the solution provider.*
* Digital Link URL
  + *This is configured automatically by the traceability driver.*
* EPCIS Query URL
  + *This is configured automatically by the traceability driver.*

For each trading party that the account will request data from allow that trading party to request data from themselves, they will add that trading party as a trading partner to the account. A trading partner is basically another account that exchanges data with the account.

## Trading Partner

For each trading party the account exchanges data with, the account will add that trading party as a trading partner. If a trading party is added to more than one account, then two trading partner records are created, and they have separated IDs.

A trading partner consists of the following data points:

* ID
  + *This can be assigned by the solution provider or can be generated by the traceability driver internally. This ID must be unique across all trading partners managed by the traceability driver instance.*
* Account ID
  + *This is taken from the account that the trading partner is being added to.*
* PGLN
* DID
  + *This is just the public key.*
* Digital Link URL
* EPCIS Query Interface URL
* Requesting API Key
  + *The Requesting API Key is generated by the Trading Partner and given to the Account so that the Account can request data from the Trading Partner.*
* Receiving API Key
  + *The Receiving API Key is generated by the Account and given to the Trading Partner so that the Trading Partner can request data from the Account.*

## Authentication

Authentication for data requests between Accounts and Trading Partner are accomplished using the following two methods:

1. Public / Private Key
2. API Key

# Digital Link Resolver

The Digital Link resolver will store one or more links internally for EPCs, GTINs, GLNs, and PGLNs. These links will be stored internally on the Mongo DB database.

The Digital Link Resolver will originally support the following link types:

* gs1:masterData
* gs1:epcis

# EPCIS Query Interface

The EPCIS Query Interface will support a subset of the EPCIS Query Interface:

**POST:** /queries/SimpleEventQuery

And it will support the following filters:

* GE\_eventTime
* LE\_eventTime
* MATCH\_anyEPC