Getting Started with IoT Data Monetization

Steve Todd, Dell Technologies Fellow

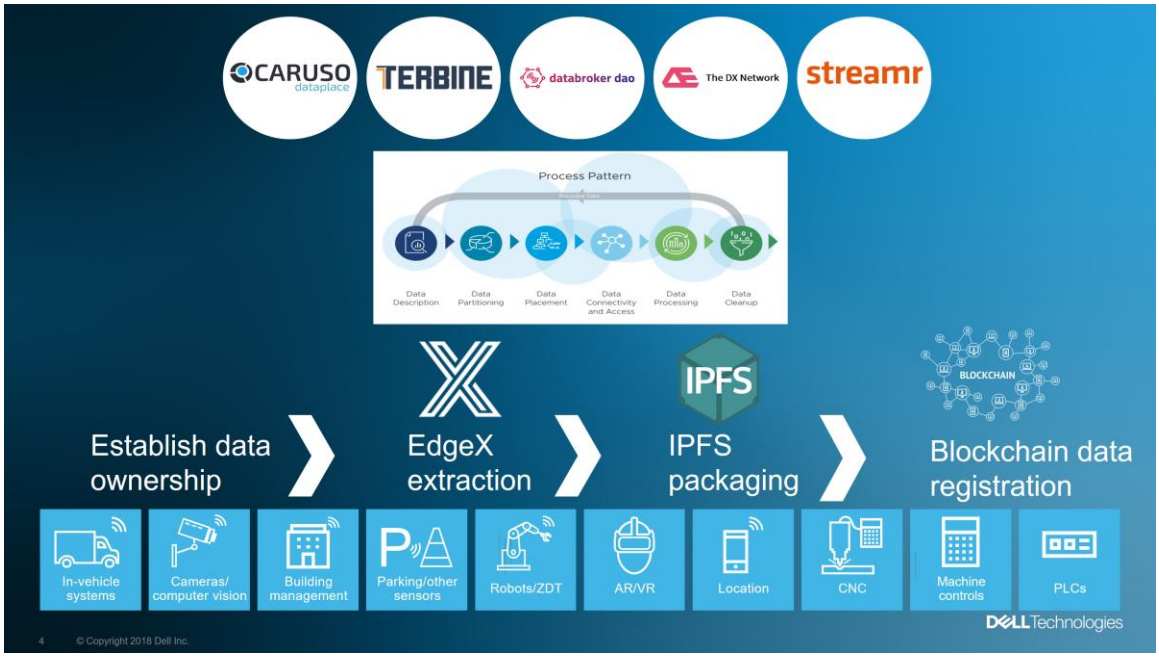


Table of Contents

Table of Figures

1 Introduction

The acronym IoT (Internet of Things) is well-known to many. Enormous numbers of embedded devices ("things" in cars, refrigerators, doorbells, street lights, etc.) are already sending their data over a network (e.g., the internet).

What is less well-known is that IoT data can also be directly monetized (e.g., exchanged for currency). This emerging revenue stream represents a significant business opportunity for IoT device vendors, corporations, and many other data-savvy start-ups.

The process of IoT data monetization, however, is fraught with complications.

The research below outlines a way for data producers to get started down the path of IoT data monetization. This paper is explicitly written for data owners: people and businesses that legally own the data that they (may someday) wish to sell.

These data producers need to become informed on how to prepare and package their data for sale, and how to best store, analyze, value, and eventually transfer data to consumers as part of a monetary transaction.

A recent paper by Gartner introduces both the opportunities and the challenges presented by the emergence of data brokers[1].

> *Treating IoT data as an asset has unlocked new opportunities for enterprises. Data and analytics leaders must explore both the opportunities and issues that arise with data monetization, including regulatory compliance, data privacy and ownership, and data ethics.*

This paper explores the emergence of data marketplaces and presents an enterprise architecture for data producers to bring their corporate data to a broker. Figure 1 will assist in outlining the flow of the article.

Figure 1 – An Architecture for IoT Data Monetization

Starting at the top of the diagram, Section 2 of this paper will introduce the phenomena of IoT data brokers and provide an overview of some emerging players. Each startup mentioned herein presents interfaces for data registration and sale; how IoT data safely and securely arrives at these interfaces will be the subject of subsequent sections.

Section 3 introduces the most complex architectural component in preparing data for sale: data orchestration. IoT sensor data often must travel through multiple Operations Technology (OT) and Information Technology (IT) layers and geographies before it arrives at a data broker. This section will provide an overview of data orchestration architectural requirements in the context of data monetization.

Section 4 discusses how to start building an architecture for monetizing IoT data. Coding samples illustrate a working implementation that includes currently available, open-source components. Each component will enable monetization features such as data ownership, data provenance, and data integrity. The coding samples will highlight how to get started. This implementation will serve as the production side (closer to the sensors) of a monetization strategy.

And finally, Section 5 will discuss the challenges of completing the market side (closer to the brokers) of a monetization architecture. There are functional areas of consideration that will inevitably surface when implementing architectures that are multi-cloud. The reader will be made aware of these challenges as a first step in solving them.

The first question to ask in our exploration of IoT data monetization is a simple one.

Why would any company want to purchase IoT data?

2 Industry Survey of Data Brokers

We will start with the Wikipedia definition of the Internet of Things to answer the question "What is IoT data."[2]

> *The Internet of things (IoT) is the network of devices, vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data.*

Companies are considering the purchase of IoT data because their data-hungry Artificial Intelligence (AI) algorithms don't have enough available data. This data is necessary to create increasingly better models that make better business decisions. As Gartner emphasizes, there is broad acceptance of the fact that treating data as an asset to be shared will facilitate the creation of better analytic models[3].

> *Data as an asset, or to use the Gartner term, "infonomics," has gained even greater interest from enterprises with the emergence of IoT and IoT data. As more connected devices generate data, data and analytics leaders are exploring, on behalf of their organizations, the possibilities and opportunities from sharing data and, ultimately, developing viable data monetization strategies.*

Figure 2 depicts five emerging data brokers. The leftmost three brokers (Caruso, Terbine, and DataBroker DAO) are described in the Gartner "Cool Vendors" report. The rightmost two companies (DX Network and Streamr) represent companies not mentioned by Gartner.

Figure 2 - Emerging Data Brokerage Companies

All five companies present different value propositions and business models. Highlighted below is a description of each company and their unique approach to brokering IoT data.

2.1 Caruso

Caruso is a German company focused on brokering connected car data. They market themselves as "The Marketplace for the Mobility Ecosystem."[4] One important goal of their data brokerage is the enablement of a "connected automotive after-market."[5]

> *Everything we do, we do to enable the players in the connective automotive aftermarket to grow their sales, profits, and brand value.*

Figure 3 illustrates a simple example showing a producer and consumer interacting in the Caruso Dataplace (in this diagram, the data from the car has already been uploaded to a data provider).

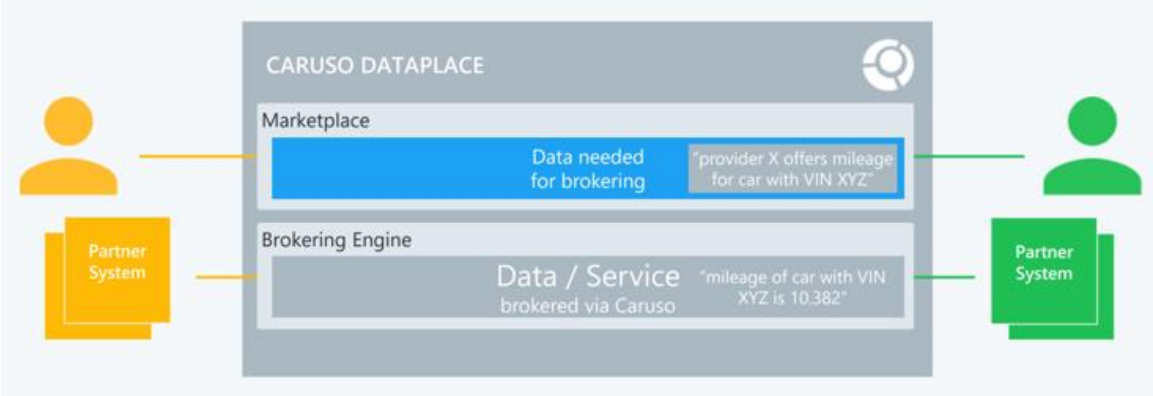


Figure 3 – Caruso Dataplace Interaction

The use case above shows two partners exchanging mileage information for a specific vehicle. Two elements characterize the architecture of the Caruso Dataplace:[6]

1. It is a marketplace that allows buyers and sellers to find one another and subsequently assist them in performing data transactions.
2. It supports a brokering engine that handles important tasks like data ingest and streaming, as well as transactional billing and logging.

Producers of connected car data, therefore, must orchestrate the telemetry coming from the sensors present within the vehicle, and shepherd that data to the marketplace.

2.2 Terbine

Terbine is a California-based company with an initial focus on brokering IoT data related to logistics and transportation. In June of 2018, Terbine announced a partnership with the Intelligent Transportation Society of America (ITS America) for a "first-in-kind" data exchange.[7]

> *This partnership allows businesses, governments, and researchers to solve problems that will lead to safer and more efficient mobility. Given the sheer volume of IoT data in the transportation ecosystem, providers and users alike have a crucial need for a master exchange. The Exchange, which is based on Terbine's cloud-based IoT data system, will fill this void. It supports any number and variety of IoT data types generated by public agencies, corporations, and universities.*

As part of the announcement, Terbine's CEO claimed that the partnership would focus on the curation and conveyance of data across public, commercial, and academic worlds. The company uses Figure 4 to highlight this emphasis.[8]

Figure 4 – Terbine Brokerage of IoT Data Across Multiple Parties

One of the unique value propositions of Terbine, according to Gartner, is their focus on "curation of IoT data to create a uniform labeling/naming convention for all data."[9] Gartner also cites significant Terbine capabilities (listed below).

- IoT data ingestion
- Real-time policy management (including provision for regulatory compliance)
- Application of rich metadata
- Rating the data
- Dynamic pricing
- APIs for third parties to access information
- Settlement
- A data rating mechanism based on the provenance of the data (ultimately extended to include user feedback).

2.3 DataBroker DAO

DataBroker DAO is a Belgian-based firm that is attempting to become the "eBay" or "Amazon" for IoT sensor data. Their approach differs from other firms in the sense that they are pursuing strong partnerships with network operators as well as "data processors" (companies that purchase data to enrich and resell it). Figure 5 outlines their proposed flow of IoT sensor data among multiple stakeholders.[10]

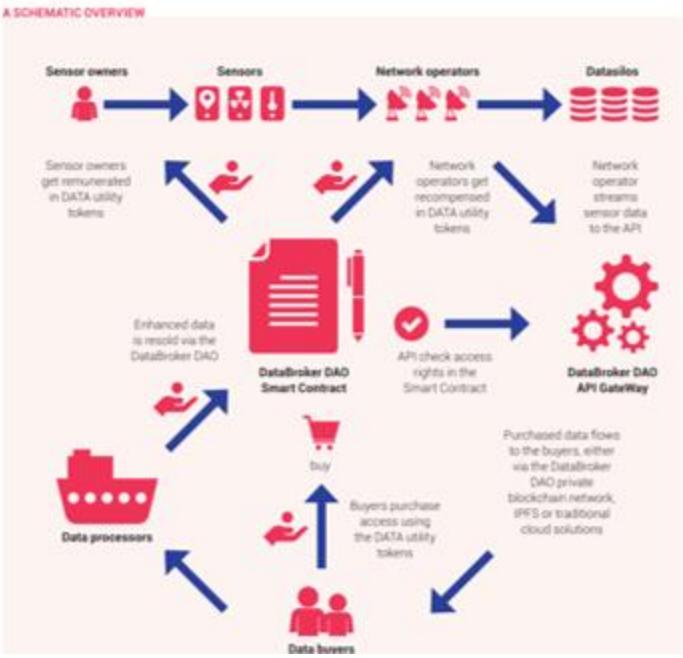


Figure 5 – DataBroker DAO IoT Sensor Monetization Workflow

DataBroker DAO is aiming to become a be-all IoT marketplace for a large number of verticals, including manufacturing, natural resources, transportation, utilities, government, smart cities, and agriculture.[11]

One glance at the DataBroker DAO architecture highlights a dependency and integration with blockchain technology:

- A token is defined (the DTX utility token) that is used by data buyers to purchase IoT data.
- Smart Contracts are central to the workflow.
- DataBroker DAO runs on the Ethereum network
- DataBroker DAO introduces the concept of dAPIs (as opposed to Ethereum dApps).

The DataBroker DAO team is upfront about their reliance on blockchain (Ethereum) technology.[12]

> *From a decentralized network perspective, [the use of the Ethereum blockchain] is also a perfect fit. Very large numbers of participants, in a trustless environment, transacting with each other is the definition of a perfect use-case.*

Their reliance on blockchain foreshadows the importance of considering internal deployment of enterprise blockchain systems to position the company for the eventual sale of IoT data assets.

2.4 DX Network

Another emerging data broker vendor that emphasizes the importance of blockchain for data marketplaces is DX Network.

The DX Network (UK-based) data marketplace is featured along with the Gartner "cool vendors" because its vision, while including IoT data, is meant to enable the brokerage of any data (e.g., beyond just sensor data).

DX Network founder Jeremiah Smith calls data marketplaces the "Holy Grail of our Information Age," because "data flow in our so-called Information Age is broken."[13]

> *In practice, a data marketplace is a piece of software which data providers and data consumers connect to through a graphical or backend interface to buy and sell data from each other.*

Smith argues that there are different types of data marketplaces: personal, business-focused, and sensor-specific. Depicted in Figure 6 are these three classifications and the characterizations which define them.[14]

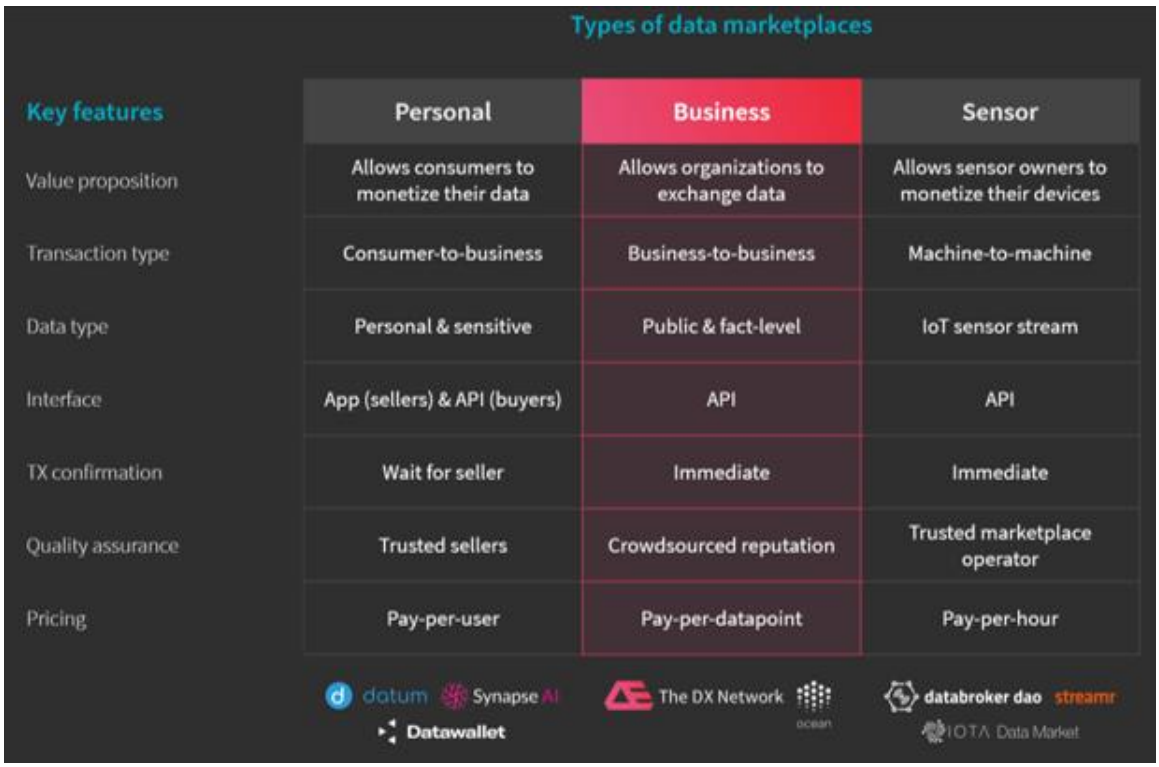
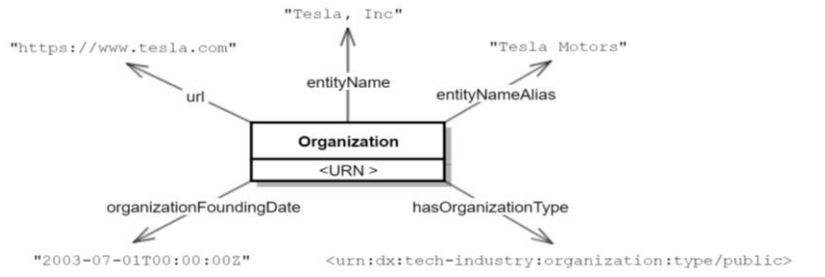


Figure 6 – Types of Data Marketplaces

A DX Network data marketplace can be created for any type or kind of data by defining a new data model or ontology. As long as the new model follows the DX Tech Industry Model, then the DX Network framework can enable data exchange (no matter what the vertical market). Figure 7 highlights a sample ontology for Tesla data.[15]



Where `<URN>` is the organization's uniform resource name which uniquely identifies Tesla, Inc. on the DX Network.

Figure 7 – DX Network Data Modeling

Once data populates into the DX Network marketplace (based on its data model), payments to the data owner can be made based on the "number of hits" achieved by that data. A DX-specific query language (DX/SPARQL) allows arbitrarily complex queries to find high-value data sets.

Founder Smith sums up the value proposition of the DX Network approach.[16]

> *Effectively, the DX Network can be thought of as a new type of knowledge sharing platform where the more utility data listed on a DX marketplace provides (number of times it gets queried by consumers), the higher the reward for the original contributor (owner). This creates direct economic incentive for data owners to list as much high-utility data as possible on DX.*

All four data marketplaces described up to this point allow a data owner to package, store, and curate data and bring it (over time) to a brokerage environment. Each broker provides tools for making the data discoverable and consistently named.

The final data marketplace is unique in its focus on the brokering of real-time, streaming IoT data.

2.5 Streamr

Streamr is a crowd-funded open source project with contributors around the world (headquartered in Switzerland), with a focus on the trading of real-time sensor feeds.[17]

> *Streamr is making these data streams tradeable. We provide a single interface for real-time data delivery and payment, using our cryptographic token, $DATA. So, if you choose, your car can sell the information it produces and buy back the information it needs. And it all happens automatically.*

Figure 8 depicts the vision for the Streamr system.[18]

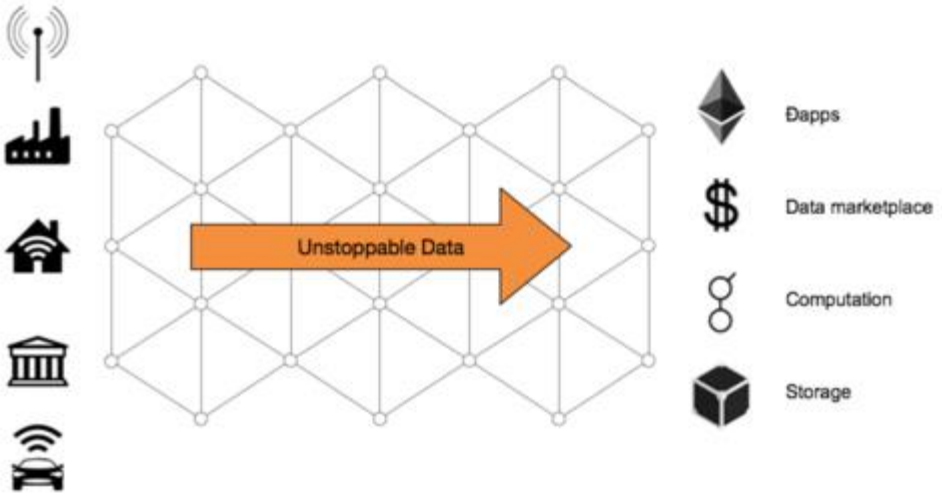


Figure 8 - Streamr Vision to Deliver Live IoT Data

Like DataBrokerDAO, Streamr is built on top of Ethereum and makes heavy use of tokens, smart contracts, and distributed applications and data transfer. Figure 9 highlights the architecture that sits on top of Ethereum.[19]

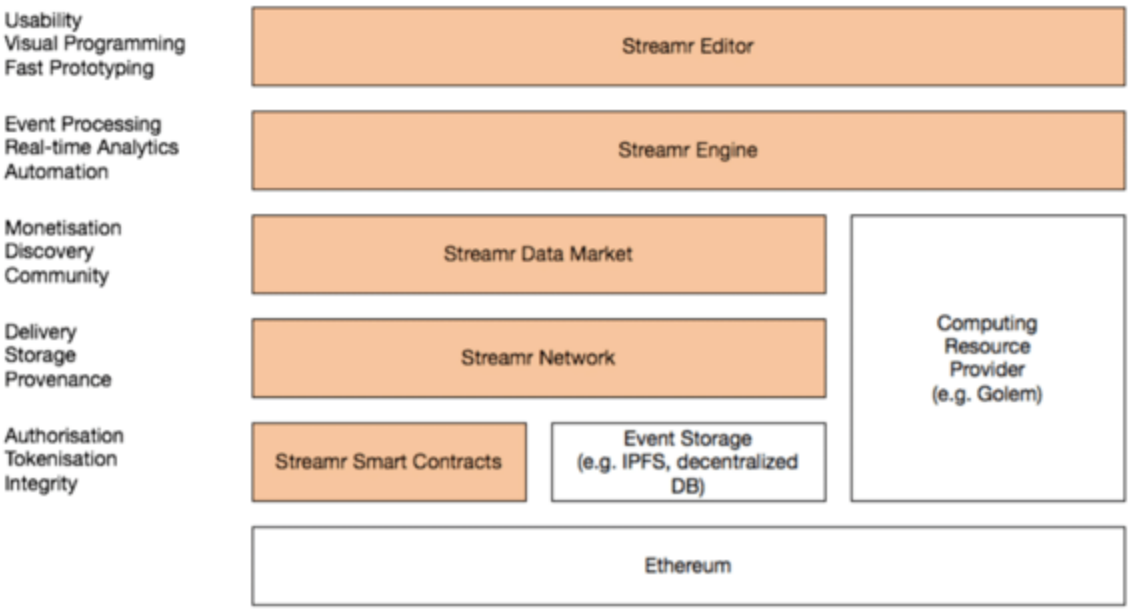


Figure 9 - Layered Stack for Streamr Network

The Streamr architecture is instructive for two reasons:

1. The bottom-most layers imply that interested sellers must possess the ability to interact with blockchains (e.g., Ethereum) and smart contracts to participate in monetization.

2. The emphasis on networking and real-time event processing/analytics means that corporate data suppliers must be positioned to provide data in a streaming form.

The five data brokers described above provide a broad variety of capabilities, architectures, and APIs.

Also, different marketplaces will expect previously-packaged data, while some marketplaces will expect live, real-time data.

In both instances, the integrity and provenance of the data will be critical.

Corporate data producers that aren't ready to plug into the heterogeneity described above may be locking themselves out of a critical new revenue source. Figure 10 highlights the challenge in its most simple form: how can we best describe and secure the flow of data from a sensor to a marketplace while maximizing data profits?

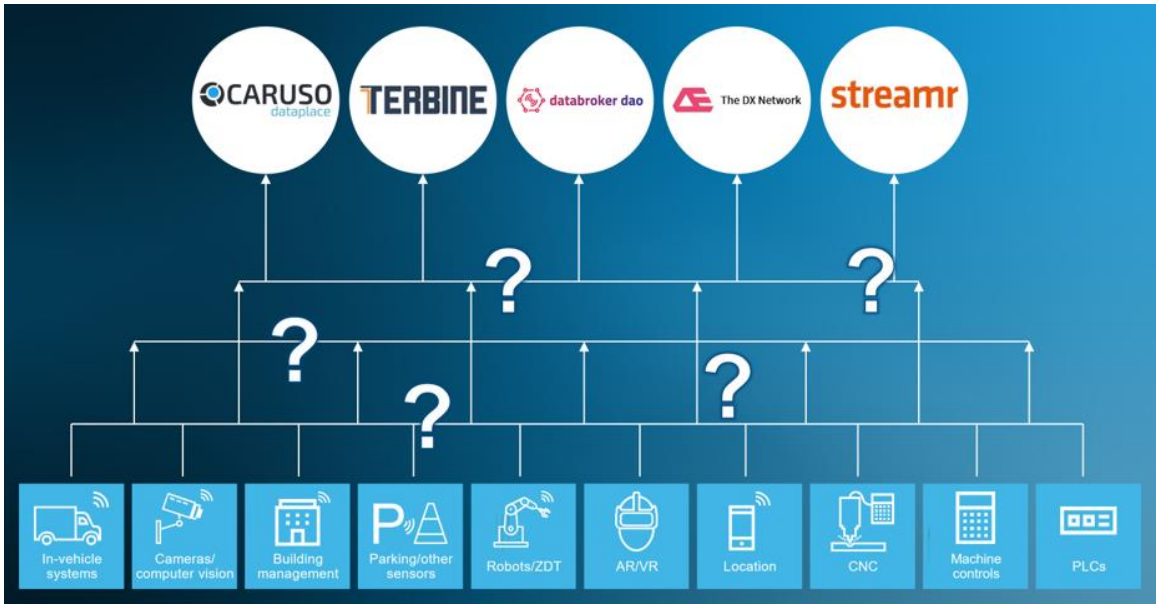


Figure 10 - Orchestrating Sensor Data Flow for Maximum Data Profits

The good news is that this heterogeneity can be scoped and managed by taking an in-depth look at common data orchestration patterns.

3 VMware Six-Sevens: Data Orchestration

The problem depicted in Figure 10 is a multi-cloud problem. Efficiently and safely controlling the flow of data from and between multiple computing environments is the domain of VMware technologists.

These technologists, as they architect multi-cloud solutions across a wide variety of use cases and domains, have noticed six common data orchestration processes that must be present to protect and deliver data in a distributed fashion. This multi-cloud pattern is

known as the Six-Sevens (six data orchestration steps across seven multi-cloud foundations). Figure 11 illustrates a diagram of the six data orchestration imperatives.[20]

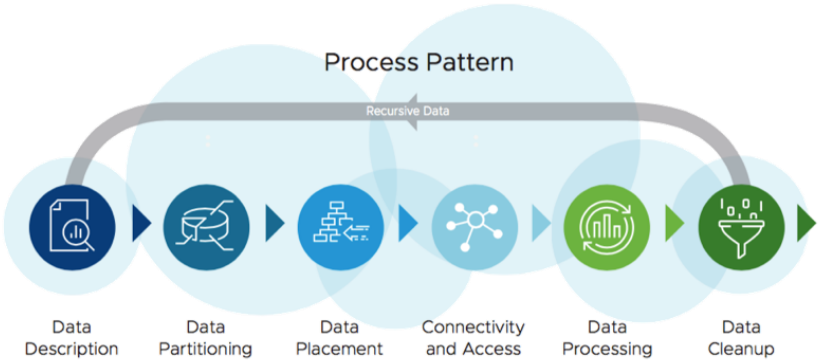


Figure 11 - VMware's Six Process Components for Data Orchestration Patterns

The pattern shown in Figure 11 applies not only to sensor data but to any data, including analytic algorithms. The Six-Sevens white paper describes AI algorithms as "executable data."

In the context of this paper, however, our consideration of data will be limited to IoT sensor data and the orchestration steps that it must undergo to arrive safely (and efficiently, with the highest data profit) to a data marketplace.

For each step in the six data orchestration processes, the sections below discuss how this step applies to the journey that IoT sensor data takes to arrive at one or more data marketplaces.

All of the steps below consider data orchestration from the data producer's perspective.

The first step is "Data Description."

3.1 Data Description

VMware describes the "Data Description" step as follows. [21]

> *Generally stated, in order to govern the use, exchange, transport, secure, or any other action on or about data, one must know the nature of data and its loss. The term "loss" as used in this context contemplates more than just mere destruction (an availability problem). Loss includes the sacrifice of any of three security objectives: availability, confidentiality, or integrity. The loss of any security objective has consequences to both internal and external organizations. Once the nature of data is understood, it can be partitioned, moved, used, and destroyed as needed for various operations as suggested by the user story.*

As a corporation builds an IoT data monetization architecture, the act of adequately describing sensor data will be foundational.

When considered against the five data marketplaces described above, there are obvious and non-obvious descriptions to associate with data.

For example, there is a wide variety of descriptive metadata that can accompany the actual sensor reading itself, including:

- The sensor ID
- The protocol over which it was emitted
- The geographic location of the sensor
- The gateway ID that initially captured the sensor data.
- The geographic location of the gateway
- The time the reading was captured
- The expected range of values or scale
- Etc.

The fact that a business may wish to monetize IoT data leads to additional descriptive items:

- Provenance data: how can we confirm the authenticity of the data?
- Data ownership: who is the original owner (and who might get paid)?
- Data quality: are the sensor readings correct?
- Data integrity: have the sensor readings been tampered with since original capture?

The presence of this type of descriptive data, at the lowest-level of sensor capture, can facilitate less friction and higher data profits as the sensor readings journey towards data brokers.

3.2 Data Partitioning

Proper data partitioning (rules governing data access) can eventually facilitate the movement of IoT data towards a data broker. VMware defines the data partitioning phase as follows:[22]

> *Data partitioning involves setting guard rails (such as governance and security rules) around which data access can or should occur. While one can generally access data through various programmatic mechanisms, properly classified data will significantly expedite and safeguard the copy process. Proper partitioning can and should occur in the process of preparing data for movement.*

VMware's reference to a "copy process," when applied to data brokers, means that access must be allowed to create a copy that can be consumed by a data buyer through a broker.

The partitioning of IoT sensor data is already critical for establishing access rules within a corporation; any data monetization architecture must explore the paradox of how restricted internal access may or may not be eventually overridden to permit monetization via external parties.

3.3 Data Placement

The initial placement of the data directly impacts the ability to eventually (or immediately) monetize IoT data through a broker. Consider the VMware definition of the data placement process.[23]

> *Once sufficiently partitioned, placement of data becomes possible. Data placement involves, among other issues, selecting the right storage, security profiles, and cloud endpoint (location) for the selected storage type. Such criteria raise issues of data sovereignty, gravity, and speed of data access. The nature of resolving these issues involves information from the data description process. Further, the placement is informed by the nature of the data and the processing it may undergo*

If there is a business advantage to immediately monetize IoT sensor data by providing a live stream to a data broker (e.g., Streamr), this capability must be enabled during the data placement phase.

Streaming is interesting from the perspective of a data producer (e.g., an enterprise). Placing data into a streaming brokerage will balance against the need also to store and analyze it internally.

3.4 Data Connectivity and Access

VMware describes this connectivity and access as a logical outgrowth of data placement.[24]

> *Placing data creates the need for connectivity. By definition, data movement requires some form of communication mechanism for its transfer. Communications produce the need for—among other things—network access and application programming interface access (such as Artificial Intelligence, or AI). Programming interface access can be called upon to create the intended storage and networks for accessing that storage.*

The data connectivity and access process, when considered in the context of data brokers, means that the data suppliers must prepare the data (and the description of that data) for transfer over external, wide-area network configurations. Preparing the data also implies protecting it based on its description (e.g., must be sent over a secure link).

It also means that data buyers will need to call data brokerage APIs to query, inspect, and ask for data. The corporate impact of third parties calling these APIs to discover internally-generated IoT data and metadata will be a new requirement for many businesses. Note that the data broker will likely execute queries from data buyers.

3.5 Data Processing

The internal analysis of data by a data supplier (and the derivative data assets produced) creates some interesting problems when considered in the context of data brokers. VMware's definition of the data processing phase states the following.[25]

> *Once data gets placed—implying that storage, connectivity, and access is setup, therefore networking exists—processing the data becomes possible. Processing involves the same issues as any software: someone writes code that needs to be executed. The nature of that execution, such as processing, requires attention to issues like containment (VM or container) and the selection of tooling for the creation, deployment, and configuration and subsequent management of that software. This gives rise to considerations of Platform as a Service (PaaS) solutions.*

There are three considerations introduced by the data processing phase:

1. Does processing of IoT sensor data inside the company potentially destroy or alter the original provenance of the data? How is provenance generated and captured for derivative assets?
2. Does the derivative output of processing IoT sensor data result in new (potentially monetizable) assets? If so, will these new assets subsequently undergo the same data description, partitioning, placement and connectivity processing that will facilitate potential inclusion in data marketplaces?
3. After data sale and subsequent processing by a data buyer, are there any controls that prohibit data resale, or are there any claims on derivative data sets?

Since software is a form of data, perhaps existing software licensing models (GPL, Apache, etc.) could apply.[26]

3.6 Data Cleanup

The final phase of data orchestration involves data cleanup. VMware describes data cleanup as follows.[27]

> *Once processing is complete, the decommissioning of cloud compute, storage, and network resources becomes necessary. Cloud operators often setup default decommissioning processes such as disk wipes and network destruction.*

The emergence of data brokers offers a new possibility for the data cleanup phase: the sale of data before cleanup. If the corporation itself no longer finds the data necessary, is there a data buyer that would?

RSA Senior Distinguished Engineer Riaz Zolfonoon points out significant security issues related to this scenario:

> *From the security perspective, when enterprise data is leaving an organization (to a broker and later to a buyer), an additional "data cleaning" step may be required to anonymize the data and remove any/all traces of identifying unique aspects. This step prevents leaking sensitive information (e.g., data coming from medical devices, sensitive industrial or enterprise assets, etc.). Also, if the data packets are signed, any cleaning/anonymization must happen before signing (or else packets need to be re-signed).*

Is it possible to squeeze more revenue out of the data before decommissioning it?

Should it be decommissioned at all if data consumers are willing to pay for it?

Is there a need to keep data so that it can be replaced if the buyer loses it?

How does data sale relate to data retention and deletion policies? If the data is scheduled to be destroyed, does that mean it can never be sold?

In this section, we've discussed VMware's six processes for data orchestration and found that it is an applicable model for a data brokerage use case. It provides a framework for thinking through the entire sensor-to-broker workflow. It also allows us to propose a first-pass architecture that considers IoT monetization from the moment of sensor data capture.

4 IoT Data Monetization: Getting Started

In Section 3 we introduced a process (the VMware Six-Sevens) to guide us through the "wiring diagram" of routing sensor data to a data broker.

Before exploring an architecture to implement this wiring, however, it's important to take a step back and consider the business benefit of interacting with data brokers in the first place: data profits.

IoT data owners are looking to positively impact their balance sheet by generating new revenue streams through the sale of data.

They want to maximize their profits, which means that IoT data owners must not only look for the highest purchase price for their data, but they must also minimize the costs involved with bringing the sensor data to market.

In other words:

IoT Data profit = IoT data revenue (from a buyer) –
IoT data cost (of bringing the data to market)

When it comes to a "wiring diagram" for IoT sensor data, should this data first travel to a public cloud before it is analyzed and sent on its way to a data broker? Or should it be kept as close to the sensor as possible?

There are growing concerns (highlighted by the survey below) that analyzing data in a public cloud increases data costs excessively (and therefore eats dramatically into IoT data profits).

Forrester Consulting surveyed 300 global IT and OT professionals in November of 2018. Survey respondents were equally distributed across one of three geographies: the US, EU, and APAC. The industries represented by these respondents spanned the gamut (e.g., retail, manufacturing, transportation, etc.), and roughly 75 percent of the companies had employee populations between 1,000 and 5,000.

Forrester found that two of the top four concerns of survey respondents revolved around the cost of storing IoT data in a public cloud. Figure 12 highlights these responses.

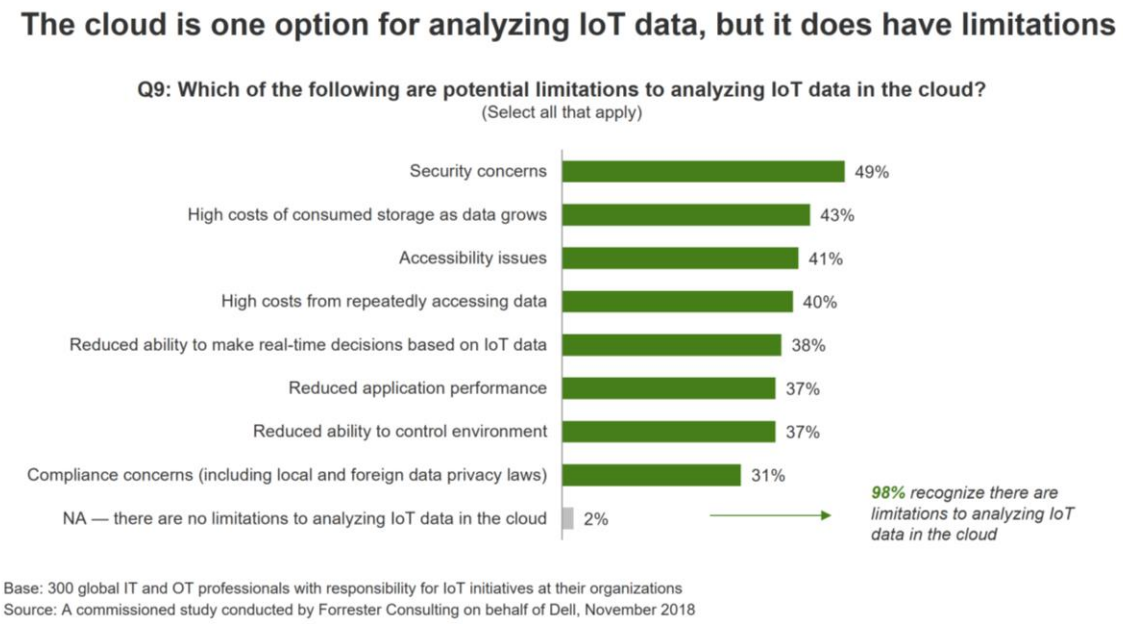


Figure 12 – IoT Industry Cost Concerns

The two cost concerns outlined in rows two and four above are as follows:

- High costs of consumed storage as data grows.
- High costs from repeatedly accessing the data.

One example commonly cited as a cost concern is public cloud egress fees.

An InfoWorld article written by David Linthicum used Figure 13 to profile the egress fees that were charged by a commonly-used cloud provider (Amazon) at the time.[28]

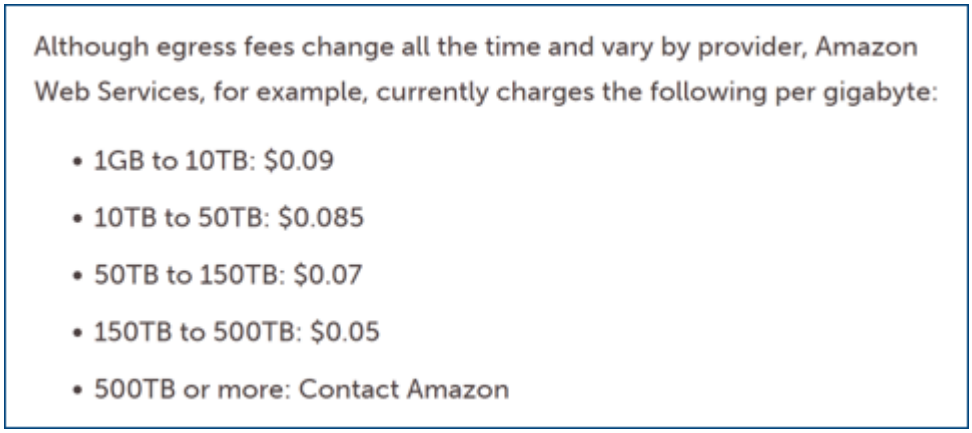


Figure 13 - Egress Fees for Moving Data from Public Clouds

If 1GB of public cloud data sells for $1.00 via a data broker, the overall data profit is reduced due to an egress fee tax of $0.09.

If that same data immediately sells to a different buyer, a similar egress tax occurs.

For this reason (and several others not related to cost), the industry is shifting towards keeping sensor data as close as possible to the sensor itself.

The Forrester survey highlights this shift by asking respondents if they are adopting a strategy of performing analytic operations as close to the sensors as possible. Nearly half of the respondents said that they were (or will be) doing so.

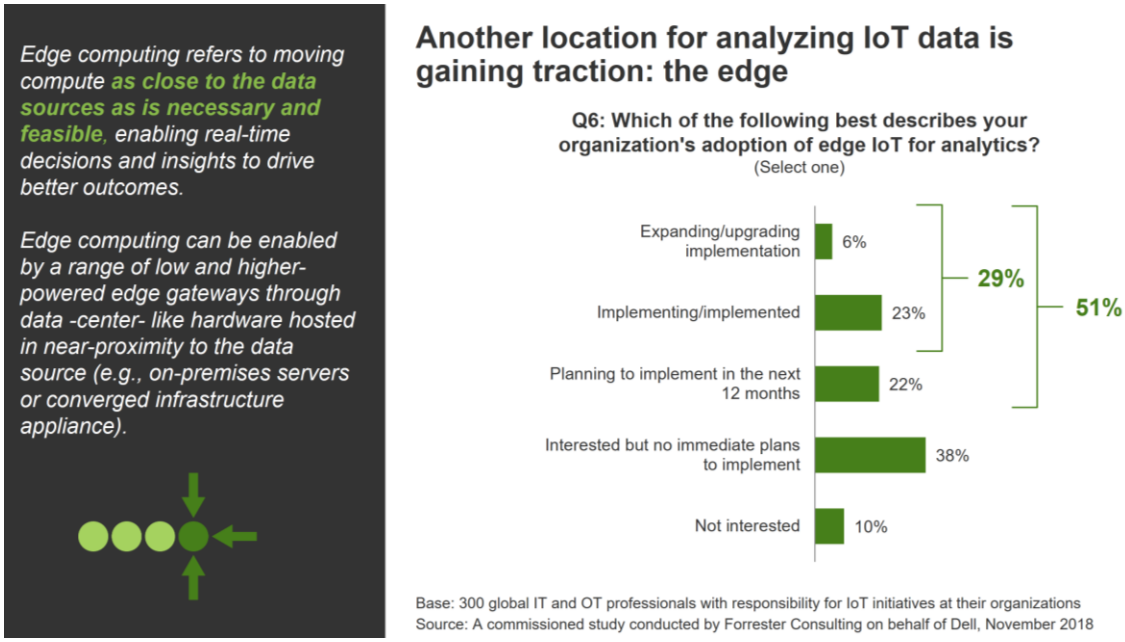


Figure 14 - Keeping Compute (and IoT data) Closer to Sensors

The survey suggests that a strategy of keeping data as close as possible to the sensors is a sound approach for controlling costs. This strategy may also, therefore, result in higher data profits.

For this reason, the approach outlined below will focus on implementing the beginning phases of the VMware Six-Seven's processes as close to the data as possible. This strategy also has governance benefits (see Section 5.2).

These initial phases will leverage open-source technologies in keeping with the theme of minimizing costs while preparing to sell data.

Discussed in the sections below are four different techniques: data ownership, EdgeX extraction, IPFS data packaging, and blockchain data registration. Figure 15 highlights these techniques.

Figure 15 - Getting Started with IoT Data Monetization

4.1 Establishing Data Ownership

The VMware Six-Sevens approach recommends the creation of a Data Description before allowing one byte of sensor data to flow into a gateway.

Two key aspects of interacting with data brokers are proof-of-ownership and accountability. Data producers that advertise valuable data for sale will benefit from the ability to provide conclusive proof that they indeed captured, validated, protected, and own the data that they are about to sell. Data buyers will likely pay higher prices for data with proven provenance (e.g., digitally signed and dated by a known owner). They will also want some recourse in cases of fraudulent data creation or possession.

Defining the provenance attributes for data is a vital Data Description activity. Chief among these attributes is data ownership. Consider the extensive list of potential "owners" of IoT sensor data:

- The sensor manufacturer
- The sensor owner
- The gateway manufacturer
- The gateway owner
- An employee identity
- A department identity
- A corporate identity
- A public (external) identity

When deciding on an identity to associate with (potentially monetizable) IoT data, it is important to remember that a data purchaser will likely perform a cryptocurrency transfer to the seller's wallet.

This wallet implies some sort of association, direct or indirect, between the identity of the data owner and the identity used to sell the data.

In theory, it would be easier if these identities were the same. Keep in mind that a data owner may wish to sell data across multiple different data marketplaces; trying to balance the mapping between internal data owners and multiple identities for data marketplaces is a non-starter.

For this reason, it is a sound strategy to explore public identities used for both purposes: registering initial data ownership and accepting payment for eventual data sale.

It follows that this identity must be decentralized; it must be valid across multiple data marketplaces (as opposed to creating new accounts for each market).

RSA is a security division within Dell Technologies with a long history of providing identity management solutions. Their latest project in this area, Project Sif, highlights the benefits of creating and managing a decentralized identity.[29]

> *A decentralized identity is a digital identity an individual creates, owns, and controls without requiring the involvement of any centralized 3rd party. Decentralized identities are accessible to everyone and designed with privacy in mind. There are no passwords and no centralized repositories of identity data.*
>
> *Through Project Sif, RSA Labs is prototyping an Identity Wallet mobile app to allow you to manage your decentralized identities. This includes creating a new decentralized identity backed by a public/private key pair. The public key is stored in a public blockchain where it can be accessed and verified by anyone.*

An ontology for decentralized identifiers (DIDs) has been officially added to the W3C Credentials Community Group (CCG). A high-level description of the proposed DID architecture will allow potential data sellers to understand this new form of identity[30].

- DIDs are unique IDs applicable to people, organizations, or devices.
- DIDs are associated with a public/private key pair and are registered on a ledger (known as a DID-registry).
- During registration, DIDs are bound to identity information contained in a DID-document. The DID-document includes the public key, supported authentication methods, and other identifying information.

In the model described below, some form of private key is used to sign sensor data digitally. The data is then associated with the owner (e.g., the gateway, or a corporation) of that private key, and the owner name and the data are distributed with the signature.

If that owner is implemented as a DID, a data purchaser can resolve the DID through the DID-registry, verify the signature, and confirm the integrity of the data. Conveniently, that DID can also be directly associated with a cryptocurrency account to receive payment.

DIDs are an excellent “getting started” strategy for IoT data monetization. However, early implementors may find the technology immature and may choose to implement a different data ownership model.

Regardless of choice, the decision of ownership must be made. Once an identity is confirmed, and the data provenance information identified and described, the ingestion process can begin.

Before it begins, however, the sensor ingest process must be configured to assist with the next step in the Six-Sevens process: Data Partitioning.

4.2 EdgeX Foundry Extraction

EdgeX Foundry is "described as a community-forged, open platform for the IoT edge."[31]

> *Vendor-neutral, open source, loosely-coupled microservices framework providing you the choice to plug and play from a growing ecosystem of available 3rd party offerings or augment with your own proprietary innovations. With a focus on the IoT Edge, EdgeX simplifies the process to Design, Develop and Deploy solutions across industrial, enterprise, and consumer applications.*

Figure 16 depicts the high-level architecture of EdgeX Foundry.[32]

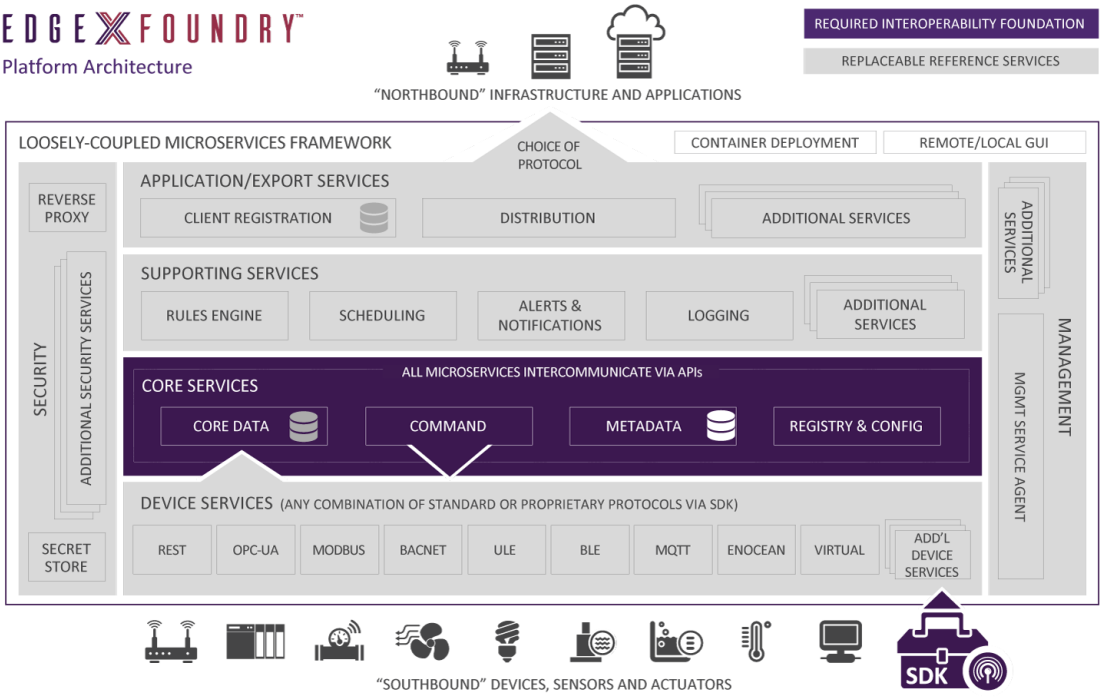


Figure 16 - EdgeX Foundry Architecture

Figure 16 shows an open-source, highly flexible (microservices-based) platform for implementing business logic as close to the IoT sensor data as possible. The southbound interface of EdgeX Foundry (labeled as "DEVICE SERVICES") sits directly on top of the data-emitting sensors. The layering provides EdgeX Foundry the unique opportunity to establish data ownership and capture data provenance at the lowest level of the data monetization pipeline.

To attach descriptive metadata (i.e., the first process in the VMware Six-Seven's framework) to incoming sensor data, an administrator must create those descriptive fields. Descriptive metadata related to incoming sensor data is packaged in a device profile. For example, a given device service that expects to connect to and ingest readings from an MQTT device will post the device profile to the EdgeX Foundry "Core Metadata" service. An actual sensor then attempts to connect to the device service. If the sensor matches the profile, its readings will be packaged in a generic data structure

within EdgeX Foundry called an “event.” This event contains one or more readings. Each reading aligns with a discrete sensor measurement and is mapped via the device profile metadata through what are called “value descriptors.” Events are then posted to the EdgeX Foundry “Core Data” microservice.

For example, as part of the data description process, it may be imperative to associate GPS information to sensor readings. Figure 17 highlights an example call to the core-data API for creating a “longitude” descriptor. This call occurs as part of an HTTP POST operation.[33]

Figure 17 - Example Method for Adding EdgeX Data Descriptions

With the provenance fields defined, the device service (the piece of software that speaks to the native protocols like MODBUS and BACNET) can also fill in longitude, latitude, and any other provenance-related fields.

It is also possible for the device service to insert an identity, fetch the identity’s private key, and digitally sign the data. Note that this step can also occur higher up in the stack (e.g., as part of a pipelining process in export services, or as part of another microservice that is running within the EdgeX architecture).

Finally, it should be noted that EdgeX Foundry may wish to batch many sensor readings together (e.g., every ten readings, or every hour) before attaching provenance data and digitally signing the entire batch. Figure 18 highlights a batched approach with grouped sensor readings (green box) with provenance data (yellow box with text, factory, and gateway provenance info) and digitally signed (the key icon).

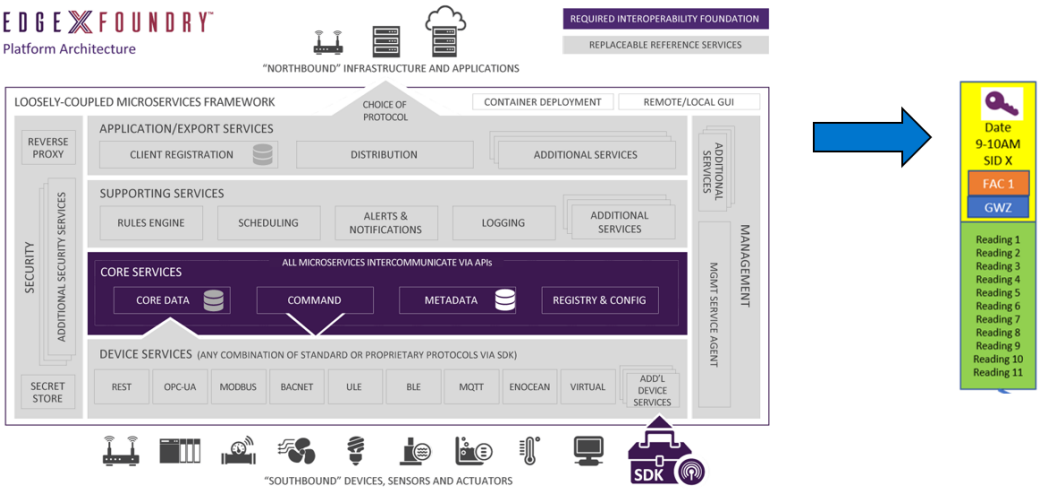


Figure 18 - Example of EdgeX Creation of Batched Provenance Packages

Note that the public key infrastructure (PKI) and code signing infrastructure is not described above. Both, however, are important considerations for a production deployment.

Now that EdgeX Foundry has assigned provenance metadata and established ownership via digital signatures, it is time to consider subsequent processes for data orchestration: Data Partitioning and Placement.

4.3 IPFS Packaging

The next open-source technology that will assist in preparing data to be monetized is the Interplanetary File System (IPFS). IPFS is a content-addressable, decentralized, object-based storage system. Data written to IPFS immediately gets hashed, and the new object is assigned the cryptographically-generated hash ID as an address.

IPFS has several benefits in a data monetization context:

- Once stored, overwriting the data can never occur (e.g., a modification creates a new object).
- Since the object ID is a hash of the content, a data purchaser (e.g., someone who buys the content in a data marketplace) can verify that no tampering occurred since creation.
- Multiple IPFS instances across multiple geographies unite into a shared object store that uses BitTorrent-style P2P communication to share objects across distributed locations. Any given EdgeX gateway, for example, can locally store content to IPFS and it will automatically be available (if desired) along with similarly generated IoT sensor data from other facilities.

Storing data to an IPFS portal is straightforward. The code required to open an IPFS connection and send packaged data to it is shown in Figure 19:

```
type restIpfsProxy struct {
        context   context.Context
        endpoint endpoint.Endpoint
        url       string
        key       string
}
```

```
func (p *restIpfsProxy) AddEvent(event contracts.Event) (proxyResponse, error) {
        response, err := p.endpoint(p.context, event)
        if err != nil {
                return proxyResponse{}, err
        }

        t := response.(proxyResponse)
        return t, nil
}
```

Figure 19 - IPFS Code Example

The sample on the left shows the structure that must be initialized by providing the endpoint URL of the IPFS instance (e.g., the network port that IPFS is listening to).

The code on the right highlights an event (the “Data Description” structure filled out by EdgeX during the sensor ingestion) being written to the IPFS endpoint, and the response is returned. Note that in the VMware example below the IPFS hash value will be retrieved (and stored in a ledger).

Once data is safely written, IPFS uses peer-to-peer messaging techniques that allow multiple IPFS nodes to share a namespace across geographies (e.g., a set of factories located in different regions). Figure 20 highlights five factories storing batched sensor readings into a scalable IPFS object store.

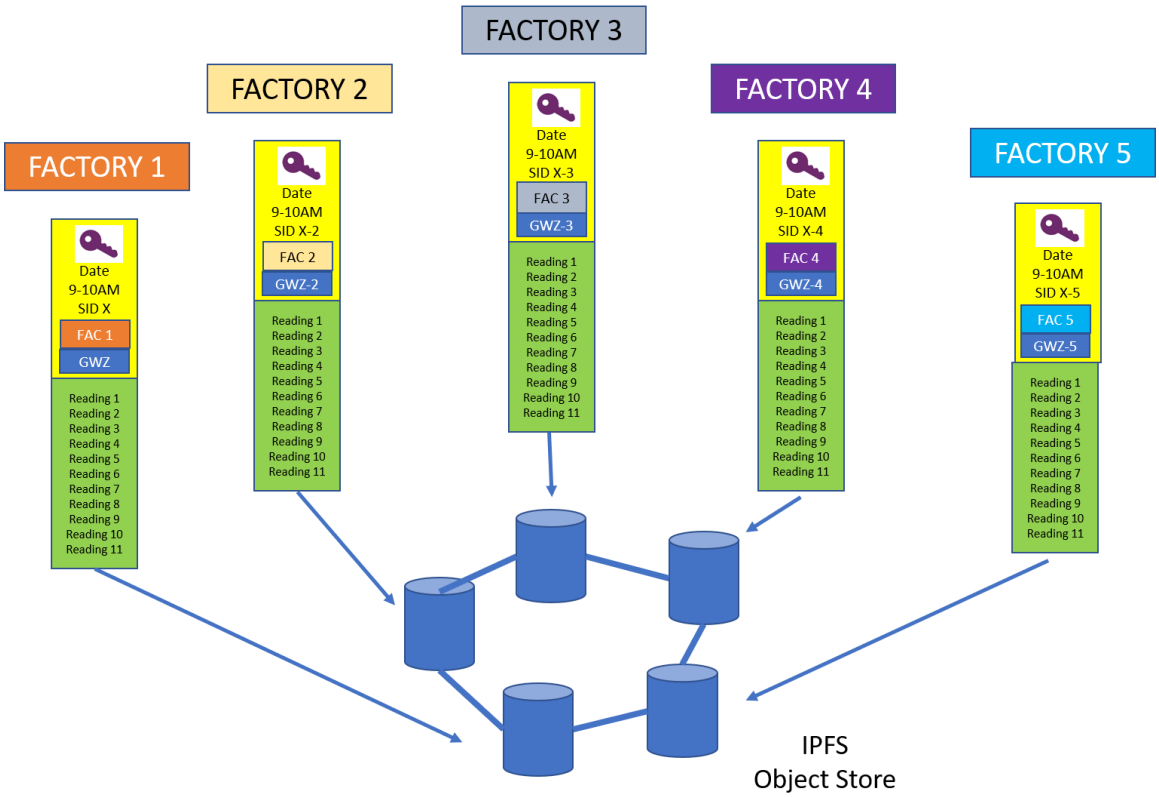


Figure 20 - Multi-Site IoT Data and Provenance Stored to IPFS

Based on the description of the data, this scaled-out IPFS object store can be the subject of constraints that limit accessibility and connectivity to this data. For example, if the Operations Technology (OT)[34] operators across the factory floor wish to deny Information Technology (IT) access to confidential production data, this would allow only a limited set of departments and employees to run analytics against the data.

Figure 21 highlights a scenario in which only OT developers have access to a distributed platform-as-a-service layer (e.g., a Pivotal Cloud Foundry PaaS layer) and deploy industrial applications that access the packaged IPFS data.

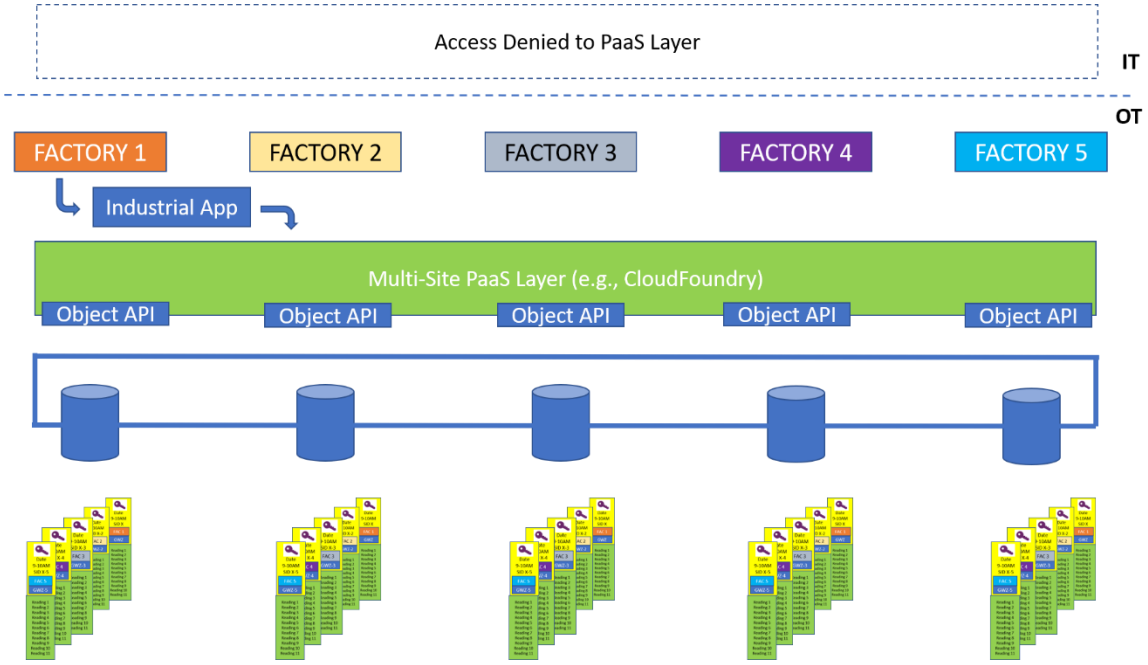


Figure 21 - Restricted (OT-only) Access to Sensor Data

Partitioning the data this way (only making it available to OT) may seem at odds with a strategy that advertises data to external data brokers. There are two points worth making, however.

1. This architecture is helpful for OT environments in which direct access to IoT sensors can bring a halt to a factory floor.
2. This architecture can be "bridged" to an IT environment (or an external data broker) in a controlled way.

There are, however, a set of questions associated with Figure 21.

- How can local OT programmers "see" new data in other facilities/locations?
- How can they search across the entirety of the object store and find specific objects?
- Why should analytic algorithms have to parse through all of the metadata and keys (the yellow portions)? What if they only want the file to contain sensor readings?

One answer is to take the yellow metadata and duplicate it into a scalable time series store like Cassandra. If duplication is not an attractive option, another thought is to remove the metadata from IPFS entirely and store it in something like Cassandra.

However, Cassandra is not tamperproof, and it doesn't necessarily support digital signatures that conclusively prove that it was the data owner that created the metadata.

For this reason, the industry is beginning to look to blockchain as a scalable method of data registration. While blockchain may (currently) be a poorly-performing database, it does satisfy the requirements of sharing permanent records of data ownership across a corporation.

Our next section will explore the concept of using a blockchain to register IPFS entries within the enterprise, and how the approach is synergistic with data broker interactions.

4.4 Blockchain Data Registration

Figure 22 shows a different method for EdgeX partitioning of IoT sensor data, metadata, and digital signatures.

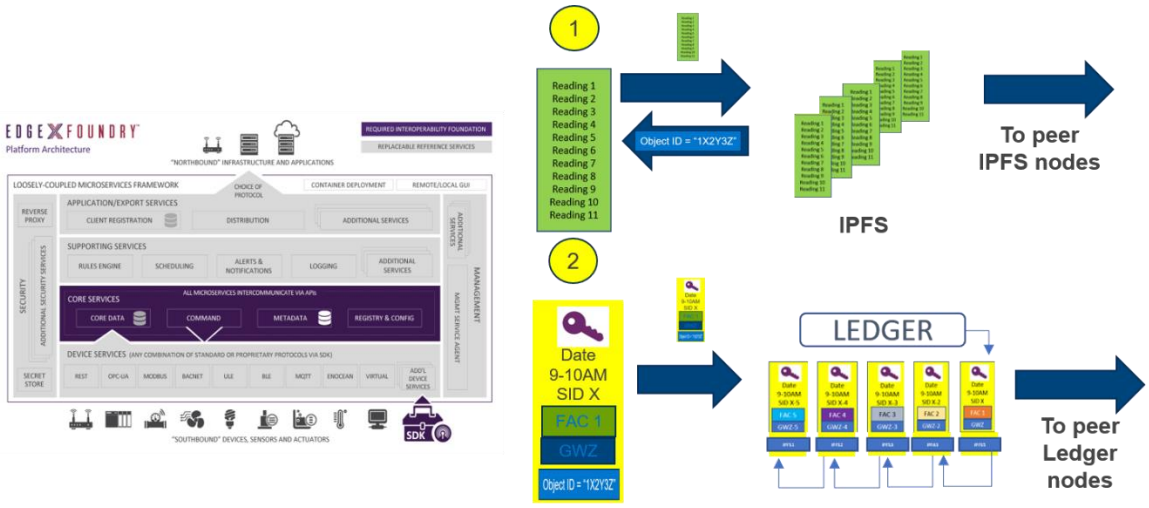


Figure 22 - EdgeX Partitioning via Blockchain

Step1 highlights EdgeX maintaining a strict separation of the data (IoT sensor readings packaged and written to IPFS). A unique object ID (1X2Y3Z) returns to EdgeX as a result. This object ID not only proves lack of tampering since the original capture, but it also is inserted into the ledger (step 2) as proof of ownership and provenance.

Note that the provenance and ownership no longer display as part of the IPFS content. The benefit is that applications can focus primarily on reading and parsing the sensor data without having to filter through other fields. It is also unlikely that a potential data purchaser would want internal metadata mixed in with actual IoT data. If desired, however, the metadata can be embedded in multiple locations.

In the example above EdgeX establishes ownership by registering the data into a blockchain (and including the IPFS Object ID as part of the transaction). The operation also includes provenance and descriptive metadata and must be signed by using the data owner's private key.

It is critical that this blockchain is a mission-critical component within the enterprise. As new data is registered, a blockchain cannot afford to perform poorly or suffer regular failures. If the corporation does not prepare for this risk (e.g., see VMware's Project Concord, described below), the existence of valuable data may not be advertised to the rest of the corporation and to marketplaces (prudent measures for minimizing this risk are described below).

Consensus-algorithm experts at VMware technology have created one of the fastest, most resilient and fault-tolerant enterprise blockchains in the industry[35]: Project Concord. Their consensus algorithm has been open-sourced[36] and interoperates with other open-source ledger technologies such as Hyperledger. Figure 23 shows how the hash (the variable "h") of the IPFS response is received and inserted into a ledger entry that includes relevant provenance metadata. This entry is then committed into the VMware blockchain.

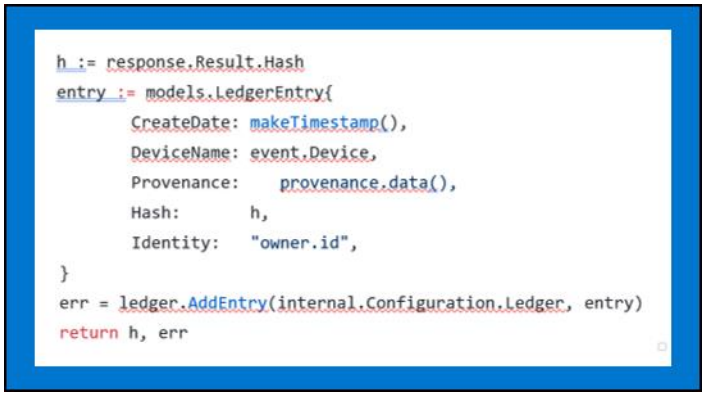


Figure 23 – Writing to VMware Ledger Coding Example

The ledger, like IPFS, broadcasts to peer locations (e.g., other factory peers in an OT environment) but it can also transmit to higher levels of the company. Indeed, external entities (e.g., data brokers) can receive transformed ledger entries that prove ownership. This exposure is not an exposure of data; it is an exposure that the data exists and that there is an owner who can make it available. The granting of access by the owner, and the sending of data to the requestor, is a complex mix of security, networking, and governance. These three attributes will be discussed in Section 5.

Figure 24 highlights sharing a ledger with IT (and eventually externally).

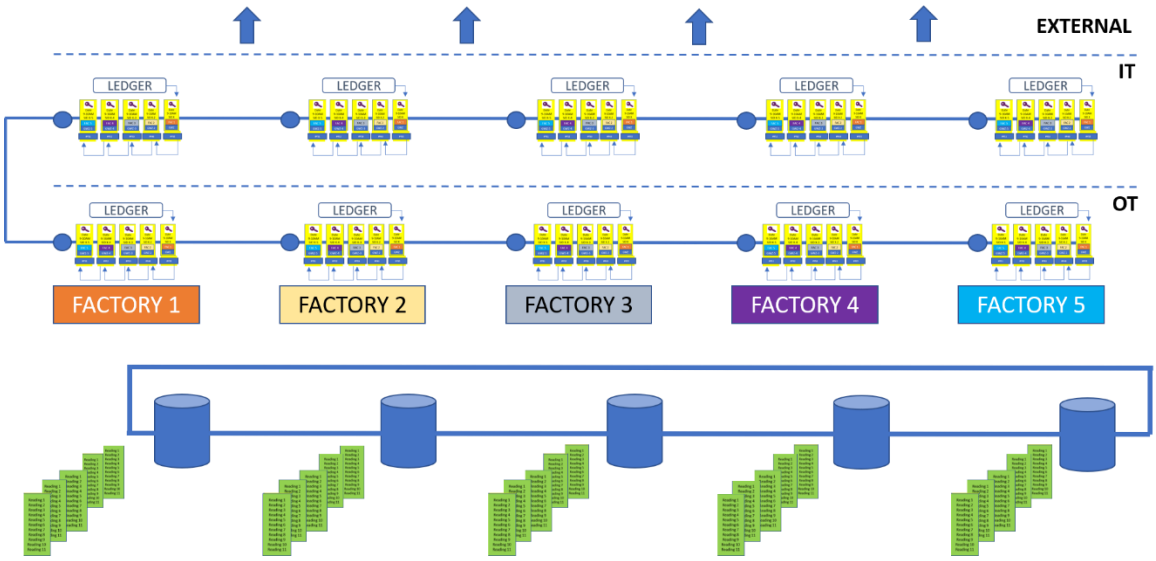


Figure 24 - Ledger architecture for multi-layer data advertisement

4.5 Getting Started Summary

In this section, we have taken the principles of the VMware Six-Seven's data orchestration framework and applied them in a practical and hands-on way.

Open-source technologies were proposed to enable experimentation and keep costs low:

- Data ownership: Key management and digital signatures must become an integral part of IoT data ingestion. New standards are emerging in the area of decentralized identities; these standards should be explored if the data owner wishes to use the same identity to sell their data. RSA's Project Sif is an excellent place to get started.
- Data extraction: EdgeX Foundry is an open framework used to highlight how provenance and data ownership attach at the earliest parts of the sensor ingest process.
- Data packaging: IPFS is an open-source object store that creates an immutable record of ingested sensor data. IPFS uses scalable techniques that permit data sharing across an organization. Ownership and provenance metadata can also be attached to the sensor data. Consumers that wish to access the data (internal or external) have an effortless way to address and consume the object (by simply using its unique object ID).
- Data registration: VMware's open blockchain consensus algorithm, Project Concord, can be built into an enterprise-class, internal blockchain implementation. Entries in an enterprise ledger can provide broader exposure to the provenance and ownership of new content (in a way that doesn't expose the content itself).

With these four building blocks in place, an organization has a set of tools to position itself for integration with emerging data monetization brokers. EdgeX Foundry provides a comprehensive and broad capability to enrich ingested IoT data from across the spectrum of IoT devices, and IPFS/blockchain offers a way to (a) establish the earliest possible provenance and trust record, (b) store the data in a scalable and immutable fashion, and (c) advertise the data to potential internal and external suitors.

This architecture is by no means a complete data monetization strategy; there are still many significant challenges to address. It is likely that corporations will have to deal with multiple data brokers across various clouds. Data exchanges and transactions will have to solve a familiar set of problems that are common to data orchestration across multi-cloud deployments.

VMware's Six-Sevens strategy will once again prove helpful in understanding and preparing to solve these challenges.

5 VMware Six-Sevens: Multi-Cloud

VMware's technologists build upon their six steps of data orchestration with the following statement: [37]

> *There are seven unavoidable concerns that require examination in order to automate the six processes of data orchestration.*

Figure 25 highlights these seven concerns as "foundational" to multi-cloud and emphasizes that these concerns intertwine with each other.

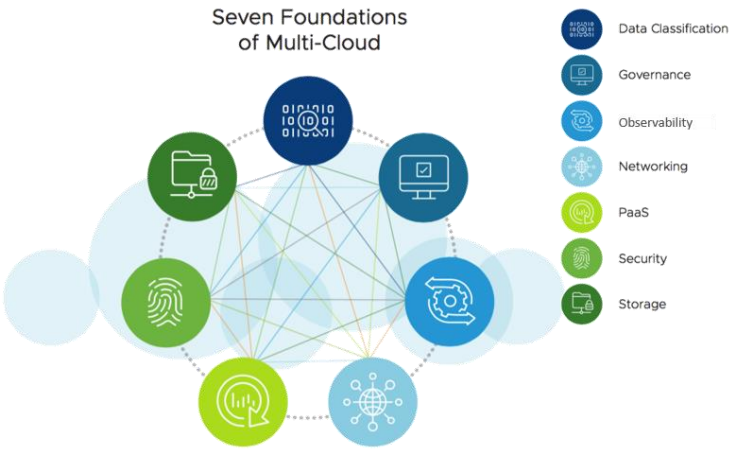


Figure 25 - VMware's Seven Foundations of Multi-Cloud

In this section, we will consider each foundation and discuss how it applies to IoT Data Monetization via data brokers.

5.1 Data Classification

The Six-Sevens framework recommends a list of data classification questions[38] that, when answered, provide significant help in the multi-cloud automation of specific data orchestration problems.

These questions are re-phrased below in the context of IoT data monetization.

- Which IoT data should be offered for sale? Which shouldn't be? Which might be?
- What are the qualities of the data to be sold? Video? Text?
- How fast is this data growing, and where should it be kept to enable quick sales transactions?
- Are there legal or policy issues that prevent a sale to specific brokers?
- Will the data be encrypted? If so, how will key management occur?
- Are there time limits on data sale or availability?
- Is the data classified (and therefore should never be sold)?

In general, consider the classification of the data and its relationship to the other six multi-cloud foundations. The Six-Sevens paper comments as follows:

> *In general, how does the description of the data influence requirements for storage, security, governance, observability, networking, or the choice of PaaS?*

5.2 Governance

One of the most critical questions about data monetization is the creation of (currently non-existent) policies that would dictate how and when data sells, how to provide record-keeping for income and exchanges, and how to perform the transfer.[39]

> *Governance is the mechanism by which organizations compose and subsequently monitor policies that control data access. Note that accessing data includes its movement, which includes accessing and passing parameters to APIs.*

Data-for-sale may be limit by data privacy laws in specific geographic regions (see the European Union GDPR and the California Consumer Privacy Act). These laws may require a data owner to describe restricted data as "not for sale."

As with the Data Classification foundation, the Governance foundation comes with a set of questions (modified here for IoT Data Monetization):

- What government, industry, or organizational policies affect processing data? Will data eventually be part of the balance sheet?
- Are there essential auditing requirements? How can every transaction be audited?
- What are the advertised service levels? Will a data purchaser immediately experience those service levels? And what are the impacts or penalties of missing them?

- How do the certificates, identities, authorization, and authentication used within a corporation (e.g., the private keys described in Section 4.1) relate to different data markets? What happens when ownership splits or merges? Or what happens if the original owner differs from the eventual seller?
- What about encrypted data? How do you make encrypted data sets available to a buyer without giving away credentials associated with the seller's infrastructure?
- How do the choices within the governance of data sale influence decisions concerning security, networking, and observability? Or backup and recovery?
- What are the risks associated with selling wrong or incomplete data, or data that shouldn't have been sold?
- What is the risk of data buyers misusing the data?

These questions have strong ties to data placement; putting IoT sensor data in a permanent, fire-walled location (e.g., a protected IPFS object store) and then limiting access is a sound start to policy enforcement. The strategy of storing data close to the sensors also steers clear of the myriad regulations that prohibit data crossing borders.

One might argue that while storing data close to the sensors helps govern against policy violations, it becomes more difficult to distribute analytics operations across a broad geography of locally-stored data. This problem can be solved by using distributed analytics frameworks like World Wide Herd (WWH). WWH implements a framework in which centralized analytic models can be distributed across decentralized, local stores, with only the results (and not the local data) crossing boundaries back to a central location.[40]

Similarly, the use of a blockchain enables audits: the recorded creation and owner of the content is accessible. It also makes sense that any data transactions (e.g., the sale of data to a broker) appear in the same (or adjacent) blockchain.

5.3 Observability

Given that there are currently no mechanisms for programmatically providing corporate IoT sensor data to emerging data brokers, there are similarly no mechanisms in place to observe the efficiency, safety, and profitability that these mechanisms are providing. The ability to observe is described as follows:[41]

> *Observability involves providing mechanisms for projecting and sensing the operating characteristics of the processes (executable data) acting on other data.*

Wherever there is currency transfer, there is fraud. IoT data monetization frameworks within a corporation must be stringently instrumented and be able to supply answers to the following questions:

- Are you monitoring for security breaches within data transactions?
- Are you selling to a buyer with which the company usually refuses to do business (e.g., a sanctioned nation, or a competitor)?
- Are data and processing compliant with the policies created as part of governance?

- Is purchased data being intercepted or re-sold, which reduces the amount of potential revenue?
- At what rate is data flowing in from IoT devices and is that rate growing? Does this indicate a need for increasing gateway counts or scaling storage needs?

As corporations build interfaces to connect to data brokers, they would be wise to instrument all components to observe the broader data monetization ecosystem.

5.4 Networking

Solving networking issues in a multi-broker environment is similar to the problems arising in a multi-cloud environment in general. Will "hot" data markets and strong buyer demand for corporate IoT data purchases saturate corporate networking devices? Will revenue be lost because of a lack of ability to sell data quickly enough? Will over-provisioning of network pipes eat into "data profits?"

VMware notes that while networking hardware will always be critical to consider, software-defined networking (SDN) is becoming a must-have. [42]

> *Networking involves creating the connectivity and access required to move data across or within clouds. This field is certainly not new, but the nature of network creation has changed over the years. With the advent of cloud-friendly and cloud-native architectures at the software level, SDN (such as VMware NSX Data Center) and SD-WAN (such as VMware NSX SD-WAN by Velocloud), multi-cloud network setup has become more dynamically driven software operations than a hardware-level, statically defined networking.*

5.5 Platform-as-a-Service (PaaS)

Interaction with data brokers means writing a whole new generation of software that forms the bridge between corporate data (e.g., IPFS data registered on a blockchain) and the marketplaces where that data sells.

New software may be needed to perform data valuation algorithms: how much can this IoT data fetch on the market? The new software will undoubtedly be required to interact with a marketplace programmatically. The Six-Sevens framework describes the need for a software delivery strategy, and this need will impact emerging IoT data monetization ecosystems:[43]

> *The processing stage of the six processes involved in multi-cloud data orchestration requires a reasonably formed software delivery strategy.*

It is indeed helpful if the same PaaS platform is used across the many variant parts of the business. For example, if Pivotal CloudFoundry is used to deploy the majority of corporate applications, it makes sense that CloudFoundry should also be used for writing new applications that interact with data marketplaces. Below are some questions to consider when it comes to the use of a PaaS in a data broker environment.

- Does the PaaS provide for facilities that expedite, help secure and implement all phases of monetization, including valuation, IoT data ingestion, or data broker integration?
- Is the PaaS platform intended for use both in the cloud and internally?
- Can the PaaS platform and application be tailored to support multi-cloud deployments easily?
- Does the PaaS have built-in observability, governance, security, and other fundamental services?

5.6 Security

For data marketplaces, the definition of Six-Seven's security is brief and to the point:

> *Security involves assuring the integrity of data.*

The brevity of the statement should not detract from the enormity of the challenge of "data integrity." Integrity can be violated if the data is not available, or if privacy was violated. Security is intertwined with the other six multi-cloud foundations. Here are a few examples that highlight the span of security considerations.

- Data classification must specify ownership and the rights of the owner.
- Governance must consider the laws that attach to the data.
- Observability must monitor against fraud, theft, and corruption of the data.
- Networking must transfer data over secure paths that cannot be breached.
- PaaS platforms must prevent malicious actors from inserting or hijacking applications that run on those platforms.
- Storage platforms (Section 5.7 below) must maintain data availability in the face of failures.

To effectively monetize IoT data, you must assure the integrity of the data from cradle-to-sale. The security-related questions to ask about IoT data monetization are identical to the question proposed by the Six-Sevens framework:

- What are the consequences of compromise, where compromise means the loss of any of the three security objectives (i.e., availability, confidentiality, and integrity)?
- What methods are necessary to detect and protect against a compromise?
- What are the policies for handling and preventing a compromise?
- In brief, what mechanisms must be in place to automate the security layer of all of the data within the multi-cloud environment?
- How does security inform storage, observability, data classification, networking, PaaS, and governance?

Keeping data close to the sensors (and preventing initial distribution) also has the side benefit of preventing unauthorized sale of the data before the actual owner has a chance to do so.

5.7 Storage

The storage of data in an IoT monetization context is another area that impacts all of the others[44]:

> *Storage is inherently a fundamental issue that addresses multiple areas: hardware, file systems (software), encryption and therefore key rotations and so on. Considerations include how storage informs security, governance, networking, and observability, and therefore how to automate those areas in order to properly utilize the storage so as to prevent any loss of the data security objectives.*

This paper has proposed a storage framework (e.g., IPFS) that lives close to the sensors. As the data flows to other internal users (e.g., the IT department) or potential external users (e.g., data brokers), will IPFS be the right choice for moving data to those new locations?

Other considerations for storage relate to how the ledger data is stored and managed. What type of blockchain will be used? What are the performance characteristics of writing to it? How are failures handled? What is the latency for confirming that the entry has been validated?

As described previously, VMware's Scalable Byzantine Fault Tolerance (SBFT) algorithm (Project Concord) is an open-source consensus framework that provides enterprise-class performance, latency, and failure handling while storing data across potentially hundreds of nodes. For example, SBFT was able to process several months' worth of Ethereum transactions and process them in a matter of hours (a 10X speed-up)[45].

One last point of discussion concerning storage is the desire to integrate with streaming data marketplaces (e.g., Streamr described in Section 2.5).

If data flows out of a sensor and into a streaming monetization marketplace, does it even need to be stored at all?

EdgeX Foundry, fortunately, can "fork" the data and immediately send it on to a streaming environment, while also packaging data as it passes by and eventually storing the batched readings into IPFS.

6 Summary

Will IoT data marketplaces emerge and become commonplace?

Based on the current shortage of high-quality IoT sensor readings for ever-increasing numbers of AI-hungry algorithms, intuition would suggest yes.

And Gartner is not pulling any punches with their predictions, suggesting that in less than five years (2023), there will be over 200 IoT data brokerage marketplaces, up from fewer than 10 in 2018.[46]

If Gartner is correct, then the time is right to start building data monetization architectures from the ground up.

But even if they are wrong, and data marketplaces never emerge, there is an ever-growing need within corporations to begin treating data as an asset and minimizing the costs associated with storing and analyzing it.

Practitioners that have already started analyzing IoT data closer to the sensors (i.e., the Edge) are already listing cost reductions as their #1 benefit (Figure 26).

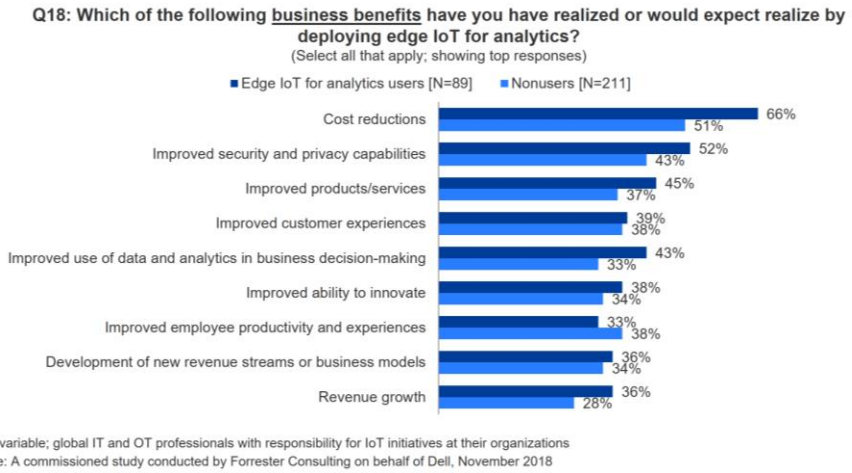


Figure 26 - Cost Benefits of Edge Storage and Analytics

There is no question that these (already-realized) cost benefits will impact the Profit-and-Loss (P&L) of the business. Therefore, implementing the principles of identity, open-data gathering (EdgeX Foundry), scalable storage, and blockchain registries can help advance this agenda.

But these principles will also favorably position a corporation for eventual integration with data brokers and open new revenue streams for data-savvy practitioners.

The concepts described in this white paper present a straightforward way to prepare for the eventuality of data marketplaces. It recommends open-source frameworks to minimize data costs and maximize data profits. For those interested in moving forward with this process, the following insights may assist in facilitating monetization.

1. Create a data ownership strategy that considers identities that are amenable to public data marketplaces. The research team from RSA is exploring decentralized identifies as part of Project Sif.
2. Collect provenance and establish ownership of IoT sensor data as early as possible. The EdgeX Foundry project is an open framework that can serve this purpose.
3. Package and store data for sale using scale-out storage frameworks like IPFS. It may be desirable to use enterprise-class systems to achieve massive scale-out, performance, and availability.
4. Register and advertise the data internally using a fast, resilient ledger that includes the capabilities of VMware's Project Concord.

5. Use VMware's Six-Sevens approach to uncover (and solve for) the specific multi-cloud issues that arise when moving data to external marketplaces.

Don't miss out on IoT data revenues! The approach described above will go a long way towards improving corporate balance sheets by treating data as an asset.

Endnotes

[1] Berthelsen, Emil, Lheureux, Benoit, Hatton, Matt, and Davenport, Jonathan. Cool Vendors in IoT Data Exchanges and Brokers. Gartner.com. August 22, 2018. https://www.gartner.com/doc/3887973.
[2] Internet of Things. Wikipedia.org. December 21, 2018. https://en.wikipedia.org/wiki/Internet_of_things.
[3] Gartner. Page 2.
[4] Caruso.com. December 12, 2018. https://www.caruso-dataplace.com/marketplace/.
[5] Naab, Mathias, and Knodel, Jens. Architecture of the Caruso Ecosystem. https://resources.sei.cmu.edu/asset_files/Presentation/2018_017_001_519117.pdf.
[6] Ibid. Page 22.
[7] Smart & Resilient Cities. ITS America and Terbine To Launch Transportation Data Exchange. June 5, 2018. https://www.smartresilient.com/its-america-terbine-data.
[8] Terbine.com. December 12, 2018. https://terbine.com/our-story/.
[9] Gartner, Page 7.
[10] Van Niekirk, Matthew, and van der Veer, Roderick. DATABROKER DAO WHITEPAPER. https://databrokerdao.com/wp-content/uploads/2018/09/whitepaper_databrokerdao.pdf.
[11] Ibid. Page 10.
[12] Ibid. Page 14.
[13] Smith, Jeremiah. Data Marketplaces: the Holy Grail of our Information Age. Hackernoon.com. July 25, 2018. https://hackernoon.com/data-marketplaces-the-holy-grail-of-our-information-age-1211a6fec390.
[14] Data Marketplaces Are the Cornerstone of the Emerging Data Economy. DX.Network.com. https://dx.network/#data-marketplaces.
[15] Smith, Jeremiah. The World's 1st Blockchain-based Business Data Marketplace: Real Use Cases [Part 1]. Medium.com. September 10, 2018.
[16] Smith, Jeremiah. The World's 1st Blockchain-based Business Data Marketplace: API Overview [Part 2]. Medium.com. October 10, 2018. https://medium.com/thedxnetwork/the-worlds-1st-blockchain-based-business-data-marketplace-api-overview-part-2-eee032ff1f56.
[17] Streamr about page. Streamr.com. December 12, 2018. https://www.streamr.com/about/.
[18] Unstoppable Data for Unstoppable Apps: DATACoin by Streamr. July 25 2017. https://s3.amazonaws.com/streamr-public/streamr-datacoin-whitepaper-2017-07-25-v1_0.pdf.
[19] Ibid.
[20] A Multi-Cloud Pattern: The Six-Sevens. VMware.com. November 2018. https://blogs.vmware.com/services-education-insights/files/2018/11/VMware-Six-Sevens.pdf.
[21] Ibid. Section 2.1.1. Page 6.
[22] Ibid. Section 2.1.2. Page 6.
[23] Ibid. Section 2.1.3. Page 6.
[24] Ibid. Section 2.1.4. Page 7.
[25] Ibid. Section 2.1.5. Page 7.
[26] Zolfonoon, Riaz. RSA Senior Distinguished Engineer.
[27] VMware.com. Section 2.1.6. Page 7.
[28] Linthicum, David. Don't get surprised by the cloud's data-egress fees. InfoWorld.com. March 30, 2018. https://www.infoworld.com/article/3266676/cloud-computing/dont-get-surprised-by-the-clouds-data-egress-fees.html.
[29] Mullins, Brian. Project Sif: Hello Decentralized World. Rsa.com. September 26, 2018. https://community.rsa.com/community/labs/blog/2018/09/26/project-sif-hello-decentralized-world.

[30] Zolfonoon, Riaz.
[31] EdgeXFoundry Home Page. EdgeXFoundry.com. December 13, 2018. https://www.edgexfoundry.org/.
[32] Ibid.
[33] EdgeXFoundry.com. December 14, 2018. https://docs.edgexfoundry.org/core-data.html.
[34] Operations technology is defined by Wikipedia as "the use of computers to monitor or alter the physical state of a system, such as the control system for a power station or the control network for a rail system". https://en.wikipedia.org/wiki/Operational_Technology.
[35] Todd, Steve. SBFT: VMware's Blockchain Consensus Algorithm. Information Playground. April 9, 2018. https://stevetodd.typepad.com/my_weblog/2018/04/vmwares-blockchain-consensus-algorithms.html.
[36] Gueta, Guy Golan. Meet Project Concord: An Open Source Decentralized Trust Infrastructure. VMware.com. August 28, 2018. https://blogs.vmware.com/opensource/2018/08/28/meet-project-concord/.
[37] VMware.com. Section 2.2. Page 7.
[38] Ibid. Section 2.2.1. Page 8.
[39] Ibid. Section 2.2.2. Page 8.
[40] Florissi, Patricia. Distributed analytics meets distributed data. CIO.com. April 25, 2017. https://www.cio.com/article/3192446/analytics/distributed-analytics-meets-distributed-data.html.
[41] VMware.com. Section 2.2.3. Page 9.
[42] Ibid. Section 2.2.4. Page 9.
[43] Ibid. Section 2.2.5. Page 10.
[44] Ibid. Section 2.2.5. Page 11.
[45] Todd, Steve. SBFT: VMware's Blockchain Consensus Algorithm. Information Playground. April 9, 2018. https://stevetodd.typepad.com/my_weblog/2018/04/vmwares-blockchain-consensus-algorithms.html.
[46] Berthelsen, Emil, Lheureux, Benoit, Hatton, Matt, and Davenport, Jonathan. Page 2.