March 2020

BOUND OBJECT HIERARCHY SERVICE

Ben Ryder

Herve Muyal

Pablo Garcia Del Valle

Frank Van Lingen

Marcelo Yannuzzi

See next page for additional authors

Inventor(s)
Ben Ryder, Herve Muyal, Pablo Garcia Del Valle, Frank Van Lingen, Marcelo Yannuzzi, Frank Michaud, and Anuj Jain

BOUND OBJECT HIERARCHY SERVICE

AUTHORS:

Ben Ryder
Herve Muyal
Pablo Garcia Del Valle
Frank Van Lingen
Marcelo Yannuzzi
Frank Michaud
Anuj Jain

ABSTRACT

Presented herein is a hierarchical representation of uniquely identified objects within the physical world, referred to as a "Bound Object Hierarchy Service." The Bound Object Hierarchy Service enables the synchronous shared view and creation of digital twins among coordinating entities. The system enables the representation and policy based access to objects, mapped from the physical world, that would typically be considered to follow a "bound within" hierarchy. The digital custody and ownership of the physical objects can autonomously change over time within the system (e.g., as an object passes along a supply chain, determined via sensors within the environment and policy based access). The policy based approach can additionally enable third parties or a limited set of the system participants to access the object hierarchy data in order to view specific elements of the tracked objects, along with the sharing of object sensor data.

DETAILED DESCRIPTION

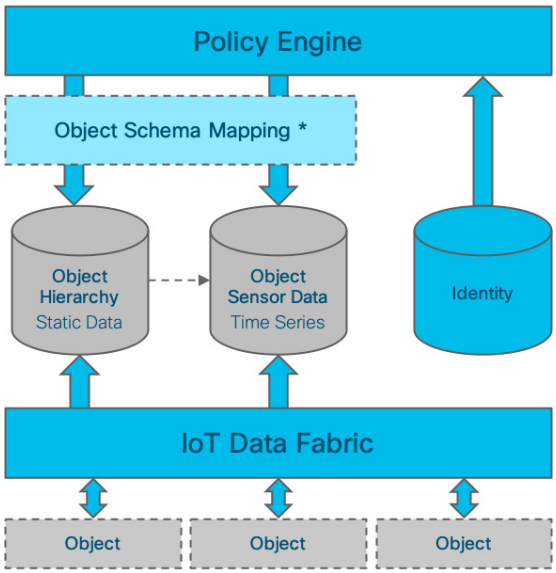
A common "real-world" problem relates to how multiple entities can synchronize a shared view of an environment in order to coordinate the secure tracking and handover of the ownership or custody of goods (i.e., asset tracking), while still allowing permissioned access to external parties. For example, future Smart Ports wish to welcome autonomous vessels in order to load or unload their cargo. However, a current challenge is how the port terminals seamlessly accept or hand over custody of containers to/from vessels at the port. Autonomous vessels, and many other use cases, are only enabled via the creation of shared digital twins, where objects and their relationships with each other are digitally tracked in a consistent way that can be accessed by all participating parties.

Presented herein is a system, referred to as a "Bound Object Hierarchy Service," that addresses the issues associated with operating, maintaining, and sharing a communal digital twin object view and data by combining several technologies, and enables the tracking and handover of object custody, e.g., along a supply chain. The techniques presented provide entities with the ability to collect, process and transport data securely, thereby enabling the vision of a digital twin to increase automation, efficiency, safety and security. For purposes of illustration, the examples presented herein focus on the use-case of ports and container management. However, it is to be appreciated that the techniques described herein are applicable in other domains, settings, and supply chain stages.

Objects within the physical world can typically be considered to follow a "bound within" hierarchy, such as: the Package is within the Container, the Container is within the Storage Location, the Storage Location is within and belongs to the Port Terminal. This hierarchy changes as physical objects are transported, for example, when the Container is loaded onto a Container Vessel the hierarchy becomes: the Package is within the Container, the Container is on board the Container Vessel, the Container Vessel belongs to the Shipping Company. Not only does the hierarchy of the physical objects change, but so does the custody of the transported objects (i.e., in the second example, the Shipping Company is now responsible for the Container).

One approach to facilitate this situation within the digital world is through a digital twin that the Port Terminal, Shipping Company, Container Company, and Package Company can all access and interact with. Additional complexity is added when critical third parties, such as the Port Authority or Customs, need to access the digital twin to monitor the Containers themselves, or when a party wishes to share data with a limited set of other participants.

This can be achieved via several steps that could be implemented independently. These steps are shown in Figure 0, below, and then described. It is proposed that a service provider could offer the identity on-boarding and management, the policy engine, object schema mapping, and integration as part of a product portfolio.



Operations that a service provider may offer are highlighted in blue, third party services in grey.

Step 1

Onboarding users and objects, and assigning data access policies, requires:

- Identity Service or Broker
- Object Hierarchy DB
- Policy Engine

Step 2

Track, trace, and access sensor data, requires:

- Identity Service or Broker
- Policy Engine
- Object Hierarchy DB
- Object Sensor Data DB

Step 3

Automated object custody exchange, requires:

- Identity Service or Broker
- Policy Engine
- Object Hierarchy DB
- Object Sensor Data DB
- IoT Data Fabric

Figure 0

Step 1) On-boarding companies (i.e., users) and their associated objects that should operate within the digital twin, each with a unique ID and current object hierarchy

- Objects can only be tracked and operate within the digital twin if the company owning the object is on-boarded into the system, and details of the object are then uploaded.
- When registering with the system, company details are captured and the company is issued an identity from the identity broker or service.
- All objects that are then registered that belong to the company are bound as “owned by” that company's unique ID.
- When the company registers an object, details are uploaded that include the object name, type, a unique identifier, and it's current position in the supply chain, i.e. the object that it is currently 'bound to', such as Container on board a Container Vessel is bound to that Container Vessel's unique ID. Data is stored in the Object Hierarchy database, which can be accessed by all participants in the system and which is restricted based on the access policies decided by the company.

- The company determines and sets the access policies for the object and the hierarchy that it falls within through the policy engine.
- If a situation arises where a company wishes to use a representation of the physical world that does not align with the static object hierarchy, this would be possible via an "object schema mapping" layer between the policy engine and the databases. This would allow different "views" on how the physical world is represented, while still utilizing the same framework and object data as partner companies in the system.

Step 2) Tracking, tracing and authorizing access to sensor data across the digital twin

- Once on-boarded, the object can be tracked by companies that have been granted access through the policy engine to the object and it's hierarchy by the company that owns the object.
- When a company has permission to view a specific object, it can see the owner of the object, the company that currently has custody of the object and the object that it is "bound to." This serves as a snap shot of the supply chain situation with regards to that object.
- If the company has permission to view objects that are also bound to the object in question, then the object's bound hierarchy can be viewed, up to where permission is granted.
- If an object is a sensor, such as a Temperature Sensor inside a Container, then access to the time series data generated by the sensor can be granted to others by the owning company through the policy engine. The time series data is stored separately from the static object hierarchy, and can be accessed via the same unique ID assigned to the sensor object within the static object hierarchy. Other information such as the GPS location of the object etc. could also be stored in the time series database, and accessed through the policy engine.

Step 3) Object custody exchange, potentially automated via sensors and an IoT data fabric.

- The "'Bound to Company" and "Bound to Object" attributes of object can change as the object moves through the supply chain (e.g., the Container being

transferred from the Port Terminal's Storage Location to the Shipping Company's Container Vessel). These changes could be mutually agreed upon by the two companies, i.e. the Port Terminal “"passes” the object to the Shipping Company, and the Shipping Company confirms the transaction, leading to the static object hierarchy to be updated.

- Where agreed and defined in the policy engine, the change could also be automated through capturing and processing the object's or the surrounding environment's sensors. For example, as the Port Terminal loads the container from the Storage Location onto the Container Ship, an RFID scanner, proximity sensors, location sensors, etc. may detect that the transaction has been completed, and trigger the system to bound object hierarchy to update.

Provided below are a selection of figures that demonstrate how the elements of the “Bound Object Hierarchy Service” might work at the data level with the same Container example as in Figure 0. In particular:

- Figure 1 illustrates how Companies might be represented (Identity) and the access policies (Policy Engine) that the Companies might have setup.
- Figure 2 shows how custody, but not ownership, of the Container might transfer between the Port Terminal and the Shipping Company, as defined in Figure 1. It is noted that FIG. 2,depicts two "Example Data Structures" referred to as “A” and “B,” representing the environment (i.e.: the state of the objects) in two different points in time. Starting from state A, an object is transferred, and ending in state B.
- Figure 3 shows examples of responses to object hierarchy queries, where the data and responses are defined based on the policies and data structures in Figures 1 and 2.
- Figure 4 shows examples of responses to object sensor data, where the data and responses are defined based on the policies and data structures in Figures 1 and 2.

Figure 1: Example Access Company List and Access Policies

Below is an example of the company identities and access policies that are followed throughout the other attached figures and examples.

List of Registered Companies

Company Unique ID (Primary Key)	Company Name
ZKnMVYGXv7	Port Terminal Company
x6z2P7lWbV	Shipping Company
N7nl9Wufku	Container Company
pdUd3dnYex	Package Company
66qOV58FJf	Port Authority & Customs

Access Policy – Shipping Company – Unique ID: pdUd3dnYex

Granted for Object (Foreign Key)	Granted to Company (Foreign Key)	Access Rights
MN0wYCTDd4	ZKnMVYGXv7	Read Bound Objects
MN0wYCTDd4	66qOV58FJf	Read Bound Objects

Access Policy – Container Company – Unique ID: N7nl9Wufku

Granted for Object (Foreign Key)	Granted to Company (Foreign Key)	Access Rights
qcwFBxh4T7	66qOV58FJf	Read Bound Objects
vNsC65MiA2	66qOV58FJf	Read Bound Objects
JjWyT1yEav	66qOV58FJf	Access Sensor Data

Access Policy – Package Company – Unique ID: pdUd3dnYex

Granted for Object (Foreign Key)	Granted to Company (Foreign Key)	Access Rights

Figure 2: Example of changing the bound object hierarchy

Below is an example of the custody of an object being transferred from one company and parent object to another. In this example, a Container Company's Container that was held within a Port Terminal's Storage Location was transferred onto a Shipping Company's Container Vessel, as such, the Port Terminal loses the object from bound object hierarchy, and it appears on the shipping company's bound object hierarchy. The data of the Container Company regarding that specific container is also updated.

The change in of a bound object hierarchy could be either submitted by the companies involved in the transaction, or automatically updated by sensor within the environment. For example, if an RFID scanner detects that a Container has been transferred onto the Container Vessel, then an IoT data fabric could be used to push a request to update the bound object hierarchy – if the policies of the companies involved permit this.

Example Data Structure A

Owned and Bound Objects – Port Terminal Company – Unique ID: ZKnMVYGXv7

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
VWlnQRG4Q8	Storage Location	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	N/A
nAxWnv4ycc	Storage Location	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	N/A
KxZ85hhOB7	Container Crane	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	nAxWnv4ycc
qcwFBxh4T7	Container	Bound to Company	N7nl9Wufku	ZKnMVYGXv7	VWlnQRG4Q8

Owned and Bound Objects – Shipping Company – Unique ID: x6z2P7lWbV

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
MN0wYCTDd4	Container Vessel	Owned by Company	x6z2P7lWbV	x6z2P7lWbV	N/A
NNlNB89Xun	Container Vessel	Owned by Company	x6z2P7lWbV	x6z2P7lWbV	N/A
vNsC65MiA2	Container	Bound to Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Owned and Bound Objects – Container Company – Unique ID: N7nl9Wufku

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
qcwFBxh4T7	Container	Owned by Company	N7nl9Wufku	ZKnMVYGXv7	VWlnQRG4Q8
vNsC65MiA2	Container	Owned by Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4
JjWyT1yEav	Temp. Sensor	Owned by Company	N7nl9Wufku	N7nl9Wufku	qcwFBxh4T7
HvWgW9R5Bt	Package	Bound to Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7
t8RBeT0FYi	Crate	Bound to Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Owned and Bound Objects – Package Company – Unique ID: pdUd3dnYex

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
HvWgW9R5Bt	Package	Owned by Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7
t8RBeT0FYi	Crate	Owned by Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Example Data Structure B

Owned and Bound Objects – Port Terminal Company – Unique ID: ZKnMVYGXv7

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
VWlnQRG4Q8	Storage Location	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	N/A
nAxWnv4ycc	Storage Location	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	N/A
KxZ85hhOB7	Container Crane	Owned by Company	ZKnMVYGXv7	ZKnMVYGXv7	nAxWnv4ycc

Owned and Bound Objects – Shipping Company – Unique ID: x6z2P7lWbV

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
MN0wYCTDd4	Container Vessel	Owned by Company	x6z2P7lWbV	x6z2P7lWbV	N/A
NNlNB89Xun	Container Vessel	Owned by Company	x6z2P7lWbV	x6z2P7lWbV	N/A
vNsC65MiA2	Container	Bound to Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4
qcwFBxh4T7	Container	Bound to Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Owned and Bound Objects – Container Company – Unique ID: N7nl9Wufku

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
qcwFBxh4T7	Container	Owned by Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4
vNsC65MiA2	Container	Owned by Company	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4
JjWyT1yEav	Temp. Sensor	Owned by Company	N7nl9Wufku	N7nl9Wufku	qcwFBxh4T7
HvWgW9R5Bt	Package	Bound to Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7
t8RBeT0FYi	Crate	Bound to Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Owned and Bound Objects – Package Company – Unique ID: pdUd3dnYex

Object Unique ID (Primary Key)	Object Type	Company Object Relationship	Owned by Company (Foreign Key)	Bound to Company (Foreign Key)	Bound to Object (Foreign Key)
HvWgW9R5Bt	Package	Owned by Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7
t8RBeT0FYi	Crate	Owned by Company	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Figure 3: Example of Policy Based Responses

Below are two examples of varying responses to an Bound Object Hierarchy Request for the same object from two different companies (following the data structure shown in Figure 2 – Example Data Structure B). The company in example A is authorized by the policies of both companies, and receives greater detailed hierarchical data than the company in example B, which is only authorized by one of the companies.

Example Request Response A

Bound Object Hierarchy Request from: Port Authority & Customs – Unique ID: 66qOV58FJf
Response for Container Vessel with Unique ID: MN0wYCTDd4

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
MN0wYCTDd4	Container Vessel	x6z2P7lWbV	x6z2P7lWbV	N/A

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
qcwFBxh4T7	Container	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
JjWyT1yEav	Temp. Sensor	N7nl9Wufku	N7nl9Wufku	qcwFBxh4T7

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
HvWgW9R5Bt	Package	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
t8RBeT0FYi	Crate	pdUd3dnYex	N7nl9Wufku	qcwFBxh4T7

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
vNsC65MiA2	Container	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Example Request Response B

Bound Object Hierarchy Request from: Port Terminal Company – Unique ID: ZKnMVYGXv7
Response for Container Vessel with Unique ID: MN0wYCTDd4

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
MN0wYCTDd4	Container Vessel	x6z2P7lWbV	x6z2P7lWbV	N/A

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
qcwFBxh4T7	Container	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Object Unique ID	Object Type	Owned by Company	Bound to Company	Bound to Object
vNsC65MiA2	Container	N7nl9Wufku	x6z2P7lWbV	MN0wYCTDd4

Figure 4: Example of Policy Based Responses

Below are two examples of varying responses to an Object Sensor Data Request for the same object from two different companies. The company in example A is allowed by the policy and receives the sensor data, the company in example B is not and receives a null response.

Example Request Response A

Object Sensor Data Request from: Port Authority & Customs – Unique ID: 66qOV58FJf
Response for Temp. Sensor with Unique ID: JjWyT1yEav

Object Unique ID	Object Type	Timestamp	Measurement Unit	Value
JjWyT1yEav	Temp. Sensor	2019-08-23T13:02:21+00:00	Degrees Celsius	16.1
JjWyT1yEav	Temp. Sensor	2019-08-23T13:02:33+00:00	Degrees Celsius	16.6
JjWyT1yEav	Temp. Sensor	2019-08-23T13:02:49+00:00	Degrees Celsius	16.5
JjWyT1yEav	Temp. Sensor	2019-08-23T13:03:04+00:00	Degrees Celsius	16.3
JjWyT1yEav	Temp. Sensor	2019-08-23T13:03:16+00:00	Degrees Celsius	16.2
JjWyT1yEav	Temp. Sensor	2019-08-23T13:03:27+00:00	Degrees Celsius	16.6

Example Request Response B

Object Sensor Data Request from: Port Terminal Company – Unique ID: ZKnMVYGXv7
Response for Temp. Sensor with Unique ID: JjWyT1yEav

Object Unique ID	Object Type	Timestamp	Measurement Unit	Value