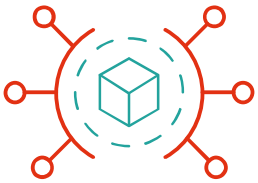

New ICT infrastructure and reference architecture to support Operations in future PI Logistics NETworks

D2.2 PI Reference Architecture – Final Version

Document Summary Information

Grant Agreement No	769119	Acronym	ICONET
Full Title	New **IC**T infrastructure and reference architecture to support **O**perations in future PI Logistics **NET**works		
Start Date	01/09/2018	Duration	30 months
Project URL	www.iconetproject.eu		
Deliverable	D2.2 PI Reference Architecture Final		
Work Package	WP2 Cloud-based PI Control and Management Platform		
Contractual due date	31/07/2020	Actual submission date	31/07/2020
Nature	R	Dissemination Level	Public
Lead Beneficiary	CLMS		
Responsible Author	Orfeas Panagou		
Contributions from	IBM, INV, VLTN, ILS, CNIT, ELU, ESC, NGS, PGBS, SON, PoA, SB		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 769119.

Disclaimer

Copyright message

Table of Contents

List of Figures

List of Tables

Glossary of terms and abbreviations used

Abbreviation / Term	Description
API	Application Programming Interface
BLE	Bluetooth Low Energy
Csv	Comma Separated Values
ERP	Enterprise Resource Planning
GPICS	General Physical Internet Case Study
GPICS	Generic Physical Internet Case Study
IMS	Inventory Management System
JSON	JavaScript Object Notation
LL	Living Lab
MES	Manufacturing Execution Systems
NFV	Network Function Virtualisation
NOLI	New Open Logistics Interconnection
NUTS	Nomenclature of Territorial Units for Statistics
OLI	Open Logistics Interconnection
PI	Physical Internet
PoA, POA	Port of Antwerp
SDK	Software Development Kit
SDN	Software Defined Networking
TEN-T	Trans-European Transport Network
TMS	Transportation Management Systems
VRPB	Vehicle Routing Problem with Backhaul
WMS	Warehouse Management System
WP	Work Package

1 Executive Summary

The key objective of this deliverable has been the production of the final blueprint of the architecture required to support PI network operations, as derived from the activities and lessons learned during the previous months of the ICONET project. The architecture presents the final definitions of the associated connectivity models, architectural modules and data structures. The final reference architecture is sufficiently generic and high-level to be widely applicable and does make use of ontologies for the definition of the relevant data structures. This report also analyses interfacing/integrating with existing logistics platforms and solutions, security and data protection, regulatory compliance and network service level monitoring aspects to ensure a viable and usable model architecture.

In this second and final version of the deliverable, the main focus was on transforming key requirements, events and data required based on a generic scenario, as well as use cases driven by the project Living Labs in a reference architecture that addresses all required capabilities. Data specifications stem from the findings of WP1 deliverables and research conducted in the development of the multiple components of the ICONET project and their interactions. The report also documents service requirements, definition of required inputs and expected outputs as well as dependencies between services, while also providing a more technical oriented approach to the potential architecture of a PI system. Furthermore, other key findings include the major events, data and decisions that need to be considered throughout the journey of a PI-container in a PI-network, as well as a blueprint for designing and implementing a PI enabled architecture in a decentralized manner, validated by the service development along with the simulated living lab scenarios.

2 Introduction

This deliverable builds upon the work described in the initial version of the architecture of ICONET and its initial analysis of the requirements, interactions and architectural elements that it described. The architecture design has evolved based on literature review on previous efforts related to Physical Internet architecture and work done on ICONET's WP1 & WP2. More specifically for WP1, the work done in T1.5 (GPICS), as well as T1.6 (PI Protocol Stack and enabling Networking Technologies) and the relevant deliverables, produced the initial guidelines and requirements for the PI architecture. The effort performed in WP2, namely T2.2 and relevant deliverables also influenced the architecture heavily (and vice-versa), as it presented the functionalities and interactions of the main services of the project.

Additionally, the challenges and goals of each Living Lab were examined in relation to the PI architecture. This report presents an architectural overview of the building blocks comprising these Living Labs, along with the PI Services that are engaged on each level.

ICONET partners also formed another basis for this architecture as they provided input during various ICONET workshops, focused on eliciting input regarding technical requirements (PI-related), potential issues and risks, and capabilities required to support PI operations, as well as restrictions and needs that came up during the development of the various services and components used. During the process of designing the initial conceptual architecture, the core modules and their interactions were also identified and mapped. In this second and final version of the reference architecture, the technical considerations and implementations that would need to be considered on a distributed PI system are further specified, resulting in the finalized decentralized Reference Architecture presented in this document.

Similar to the previous version, the layers of OLI, NOLI and the resulting composite reference model are used as a basis for defining the main architectural components, as this provides a common basis of understanding the Physical Internet concept and its' desired functionality and outcomes.

2.1 Deliverable Overview and Report Structure

This deliverable provides a conceptual reference architecture for the design and development of PI network functions and services and builds upon previous work to further expand the concepts explained therein. The deliverable starts the definition of different layers from different reference models, and their role in the PI, as they became more concrete during the previous months of the project, concluding on the representative ICONET reference model. The specifications for PI-elements and services are then described and how they fit to the previously explored concepts. The next section (Section 5) provides an example transversal scenario of a shipment from one PI node to another. Section 6 presents the design principles that led to a distributed proposed reference architecture, as well as the Living Lab considerations that the architecture addresses. The finalized blueprint of the PI Reference Architecture is also presented, shown in relation to the various building blocks as well as using an individual PI node as a basis. Its relation to decentralized networking paradigms and their design principles as they were used for the PI Reference Architecture are also presented in this section. Section 7 presents a basic ontology needed to cover the requirements of the previously mentioned concepts, along with data security considerations. Finally, the report presents the conclusions made in Section 8.

3 ICONET Conceptual Layers

The methodology followed in the previous version of this deliverable was to explain the high-level functionalities of the proposed OLI layers, on which we based the decisions made when designing the key services of the ICONET project and their interactions, which by extension heavily influenced the reference architecture. During the earlier months of the project, work done in WP1 and WP2 aimed to further specify which of these functionalities can be applied in a technical manner to enable the goals of the ICONET project. In the following section, an analysis of the high-level functionalities of OLI and NOLI layers is presented. Again, this is done on a conceptual level, as this helped tremendously to further specify which of these layers and corresponding services can truly pave the way forward for the Physical Internet concept.

3.1 OLI & NOLI layers

The output of the OLI (Open Logistics Interconnection) model (Montreuil et al., 2012) and the NOLI (New Open Logistics Interconnection) model (Colin et al., 2016) is presented in Table 2 below.

<table>
<tr><th>TCP/IP
Layer Name
(Internet)</th><th>OSI reference
Model
Layer Name</th><th>OLI
Layer Name
(Montreuil et al.)</th><th>NOLI
Layer Name
(Colin et al.)</th></tr>
<tr><td rowspan="3">Application</td><td>7. Application</td><td>7. Logistics Web</td><td>7. Product</td></tr>
<tr><td>6. Presentation</td><td>6. Encapsulation</td><td>6. Container</td></tr>
<tr><td>5. Session</td><td rowspan="2">5. Shipping</td><td>5. Order</td></tr>
<tr><td>Transport</td><td>4. Transport</td><td>4. Transport</td></tr>
<tr><td rowspan="2">Network</td><td rowspan="2">3. Network</td><td>4. Routing</td><td rowspan="2">3. Network</td></tr>
<tr><td>3. Network</td></tr>
<tr><td>Network
Access</td><td>2. Data Link</td><td>2. Link</td><td>2. Link</td></tr>
<tr><td>Physical</td><td>1. Physical</td><td>1. Physical</td><td>1.Physical
Handling</td></tr>
</table>

Table 1 TCP/IP,OSI,OLI & NOLI relation

In essence, the differences between the already examined OLI model and the OLI model is that the NOLI model attempts to further clarify and explain the functionalities of the various layers, proposing unifications and semantically important differences between layers. More specifically, the NOLI model proposes the renaming of the Logistics Web layer to Product layer, as they claim that the Physical layer definition of the OLI model would have to have definitions for all physical objects that are relevant to it(π-Products, π-Movers etc.) which is not feasible. As such, the NOLI model passes that responsibility to the entry point of the Physical Internet, which on this model is the Product layer. The product layer would be responsible for defining the products that are passed through it, and in turn, the Physical layer would only be responsible for the actual physical handling of these products, thus it is renamed to “Physical Handling” layer. Furthermore, the NOLI model proposes the

encapsulation layer to be renamed to "Container" layer, as the encapsulation of products is specified as putting products into π-Containers. Additionally, it is proposed that the Shipping Layer is broken down into Order and Transport layers respectively, to have a clearer separation of concern. Finally, the NOLI model proposes the unification of the Routing and Network layers, as their interdependency is significant, thus unifying their functions into a single conceptual layer.

Based on the considerations mentioned above, a mix of those 2 models were used as a guideline both for individual services and the overall conceptual architecture. A high-level description of the layers that the ICONET project relied upon is presented below.

3.1.1 Physical Layer

This layer monitors the physical objects of PI involved in handling and transporting cargo such as means of transport, vehicles, carriers, conveyors, stores and sorters. ICONET project investigates these concepts captured in D1.1 (PI-aligned digital and physical interconnectivity models and standards) where the solutions for generalising and functionally standardising unloading, orientation, storage and loading operations is being investigated. The Physical layer is responsible for the physical actions that need to happen for a shipment to begin and conclude its' trip throughout the π-Network based on the decisions made of the other services. As these actions already occur with a variety of different ways in the current logistics climate, further technical work on Physical layer was deemed out of scope for the ICONET project, and instead, we chose focus in providing already established outputs that can be used for already occurring Physical actions (such as picking lists for product loading etc.).

3.1.2 The Link Layer

This layer handles node to node transfer. The link layer is responsible for ensuring the smooth flow of goods between PI-nodes. To achieve that, this layer must enable the pre-evaluation of potential options, identification of potential issues across the supply chain and the suggestion of appropriate mitigation measures. In ICONET this layer provides mechanisms for efficient and reliable shipping of (sets of) PI containers from shippers to final recipients. The management of the procedures and protocols for configuring the quality of service, monitoring, verifying (acknowledgement), adjourning, terminating and diversion of shipments in an end-to-end manner is being conducted in ST2.2.3 Shipping algorithms and services will be specified in deliverable D2.4 ('PI networking, routing, shipping and encapsulation layer algorithms and services v1'), and as such, the functionality of the conceptual Link Layer will be handled by the Shipping service.

3.1.3 The Network Layer

The Network Layer is responsible for ensuring interoperability, integrity and interconnectivity between different networks. The issue of interoperability will be addressed through the definition of a common data model for all PI-operations, while interconnectivity is facilitated through the use of IoT and the definition of a protocol stack. The common data model will be constructed from the data specification of all PI-elements and all ICONET services to form the ICONET ontology, described in detail in ICONET Ontology. Specifically, in ICONET, this layer will act as a knowledge base containing information of π-Nodes, π-Hubs, routes etc. in a network.

3.1.4 The Routing Layer

The routing layer is in charge of routing the PI containers from starting point to their destination. To achieve that, the routing layer must be able to monitor the status, capability, capacity (utilization) and performance of PI

operations. ICONET will achieve this through the routing service. The routing service will be responsible for supporting optimal routing decisions, taking into account cost, environmental and operational factors such as emissions, network topology and the type of the product. The objective of this service will be to ensure the smooth operation and flow of goods between PI elements. To achieve that, the routing service requires information from the Network layer, regarding compatible locations and general information regarding the status of the PI hubs.

3.1.5 The Shipping Layer

The Shipping layer is responsible for the efficient shipping of PI container, focusing on the functional and procedural requirements, ensuring visibility of shipping status throughout the PI network. On a conceptual basis, the shipping layer will handle the propagation and processing of Order information (as part of the Order conceptual layer) and ensure visibility and quality of service of shipping (as part of the Transport conceptual layer). ICONET will address this requirement through the Shipping Service. The shipping service will be responsible for ensuring efficient and reliable shipping of PI containers from shippers to final recipients. The objective of the shipping algorithms will be to ensure quality of service and monitoring of end-to-end cargo movement. The shipping service requires data from IoT devices, regarding location and cargo status, and in conjunction with the networking service, PI-hub data in order to be able to divert shipments as needed.

3.1.6 The Encapsulation Layer

The Encapsulation Layer links products to PI containers and is responsible for the composition and decomposition of orders while maintaining visibility and traceability of PI-elements. The requirements of the encapsulation layer will be addressed through the definition of an encapsulation service that will be responsible for the efficient encapsulation assignments of products within PI containers. Encapsulation algorithms will take into account the modular load units for a product, the capacities and performance of transportation means and the status and capacities of PI hubs. In essence, the Encapsulation layer is responsible for all packing & unpacking operations that might occur in the π-Network.

3.1.7 The Logistics Web Layer

As defined in OLI, this layer provides the interfaces between PI and users. In ICONET, the responsibility of this layer is to act as an entry point of potential non-PI orders originating from legacy systems, and also provide information regarding these orders to interested parties. Additionally, integrity of PI-operations will be ensured through the use of blockchain. The role of blockchain technologies will be to ensure integrity of operations through the definition of trusted, auditable and secure distributed ledgers & smart contracts of transactions as containers flow within the PI network and with a combination of data stemming from IoT devices will be used to ensure all constraints of shipments are maintained.

3.2 Resulting ICONET services

To summarize the previous section, the

4 Service Requirements & Implementations

Having defined the services of ICONET, in the previous version of the report, the required inputs and their sources were captured. Additionally, the expected outputs, and the interaction among different services were identified, mapped and enabled. These can be found in D2.1, D2.3 and D2.4. A short overview of the simulation and optimization services is presented in the next section, while focusing on the offerings of the key 4 services. Furthermore, the IoT platform is shortly mentioned through its interactions and is further detailed in D2.7.

4.1 Simulation service specification

The previous version of this deliverable also explored the capabilities offered by the simulation service. While this enables the PI services and operations to work in tandem with simulated data, this is not part of the reference architecture. Instead, it is a good starting point to simulate the integration and features provided from the aforementioned services and provide a testing ground for the respective implementations.

To reiterate, the simulation service is an overarching service that needs to be aware of all network data in order to emulate the operations of the PI-network. Its requirements, inputs and outputs are presented in Figure 1.

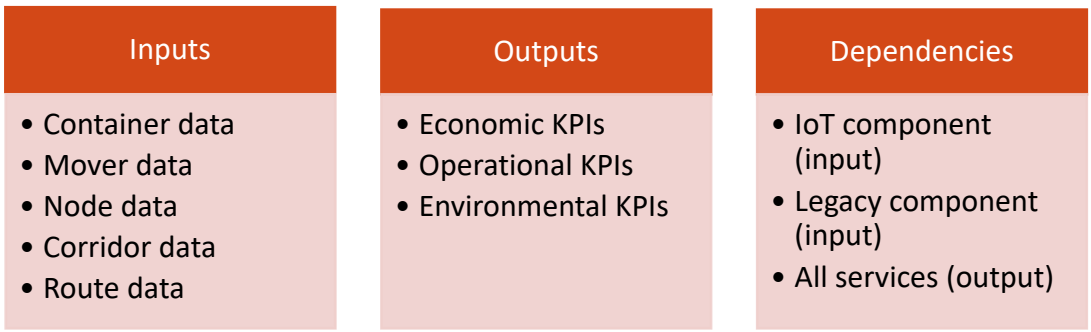

Figure 1 Simulation service requirements

The simulation service requires data from all PI-elements in order to evaluate all possible scenarios and generate the relevant KPIs that can be consumed by the rest ICONET services in order to identify the best possible network configuration. It depends on the IoT components and Legacy component for the extraction of all relevant data.

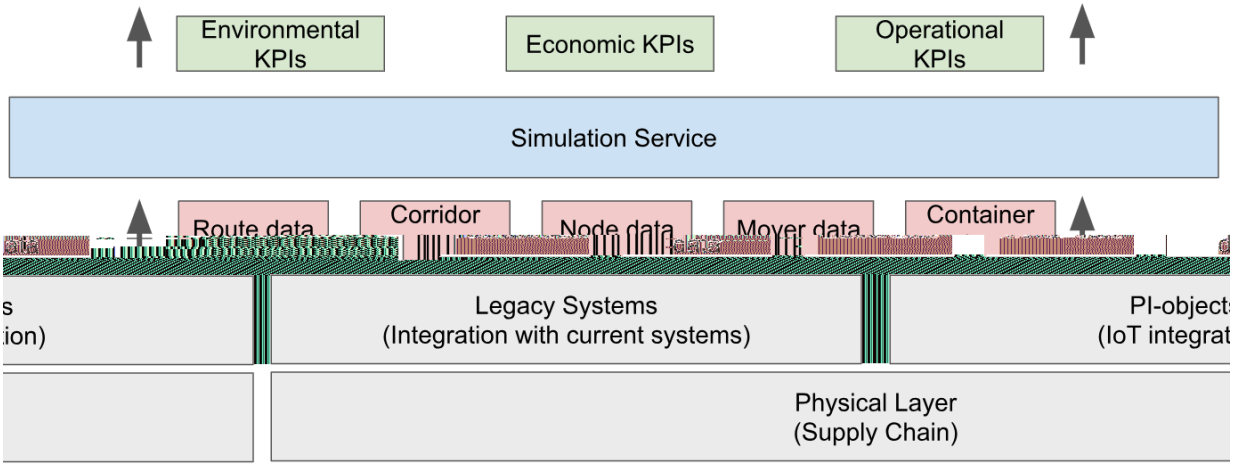

Figure 2 Flow diagram for simulation service

The diagram above represents the flow of data for the simulation service (Figure 2).

4.2 Optimisation service specification

The optimisation service is concerned specifically with optimising operations within a single PI Node. While there are a number of variations in the layout and function of a given PI Node, the operations frequently fall into similar categories:

Moving goods internally within a node
Storing goods internally within a node
Consolidation or deconsolidation of goods within a node

The optimisation service addresses all three of these through machine learning and reinforcement techniques. Due to the bespoke natures of a given PI node and the fact that machine learning approaches are highly data driven, the optimisation service is custom built per node (or node type) but uses a common framework, ensuring customisation effort is minimal. As already mentioned, data acquisition is a key requirement in using the optimisation techniques defined. The data in question can come from a number of source but most lie within the PI Node itself. While real time data is used to define the optimisation problem at hand, historical datasets power the algorithms themselves. Therefore, order history as provided by the shipping service for the node is critical, as is operation plans within the node – for example the train schedules and movements within a port, or the list of containers to be relocated on a deep-sea terminal on a given date. This data is often held within legacy IT systems outside of the PI service stack, so must be extracted. This is done by the shipping service for that node due to privacy and confidentiality concerns, but other non-confidential info could be shared via the network service as it may be consumed by neighbour nodes, depending on the data type.

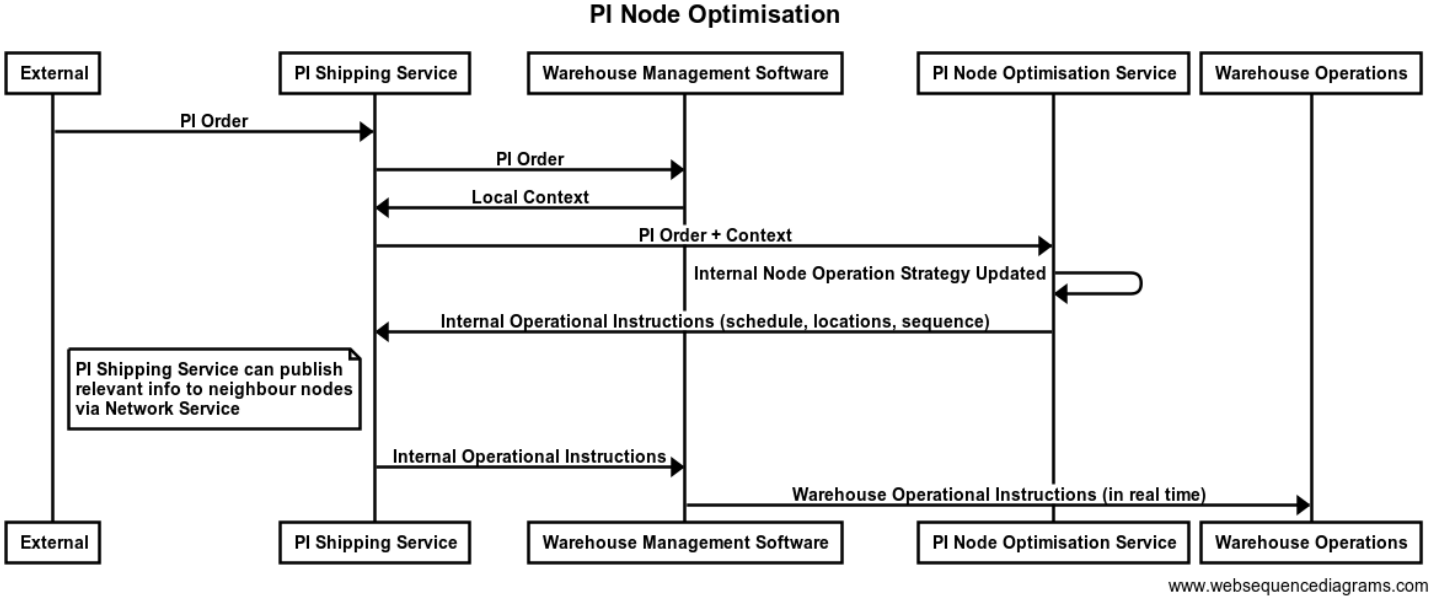

Figure 3 Optimisation Service in Context

Figure 3 Optimisation Service in Context shows the integration of the Node Optimisation Service with the PI Service Stack and local services within the PI Node. This integration represents an updated approach to the OLI service layers which has not previously included a node optimisation layer. The optimisation service is enhanced and powered by data sharing, which is a core strength of the PI concept and what ultimately powers a PI Network.

In turn, optimised PI Nodes lead results in better throughput in a PI Network. The inputs and outputs for the PI Node Optimisation Service depend on the specific optimisation problem at hand. For ICONET the focus was placed on train loading and consolidation within a port, container storage within a port and goods storage within a warehouse. The inputs in these circumstances would include train schedules for the day, Rail undertaking bookings for the day, container arrivals and departures for the day and deep-sea vessel arrival and departs for the day. In addition to port operations, warehouse operations were addressed by the optimisation service so inputs in this scenario are warehouse status (available slots, available forklifts etc.) and movement instructions (container will arrive at a certain day and time and leave at a certain day and time). The output from the PI Node Optimisation service will be an optimised plan (A loading plan, a storage plan, a train schedule, etc.) This is also summed up in Figure 4.

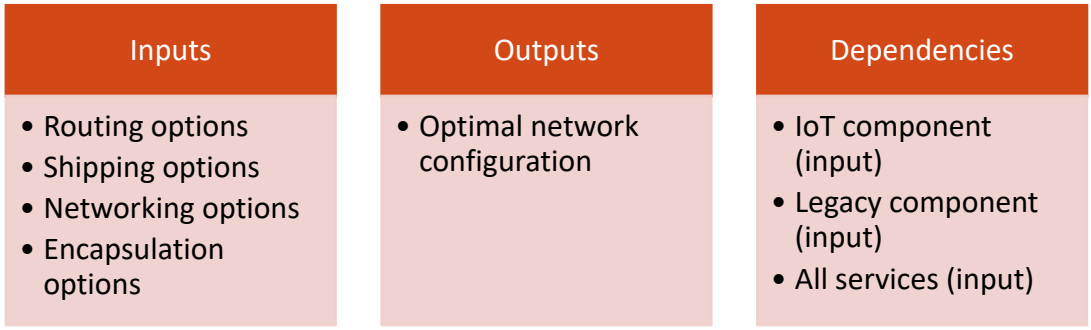

Figure 4 Optimisation service specification

4.3 Shipping, Encapsulation, Routing, and Networking service specification

The services that were identified in section 3.2 as key enablers for a PI network are described in this section. Their offerings and interactions are what allow the PI concept and as a by-product, its' conceptual architecture to be successfully encompass the functionalities and aspirations of the PI. As these services are described in length in deliverables D2.3, D2.4 and D2.5 they will examined briefly, to provide an overarching idea of the key components of the architecture.

4.3.1 Shipping Service

As mentioned before, the Shipping Service encapsulates the functions of the conceptual Shipping layer, which is further conceptually divided into the Order and Transport layers in the ICONET reference model. Its role in a PI enabled network environment, as per the DoA, is to "enable the efficient and reliable shipping of (sets of) PI containers from shippers to final recipients. Study the management of the procedures and protocols for configuring the quality of service, monitoring, verifying (acknowledgement), adjourning, terminating and diversion of shipments in an end-to-end manner, leveraging the IoT means of T2.3 in concert with the Blockchain principles of T2.4 wherever possible." In order to fulfil these goals, the Shipping service takes on the role of the overall orchestrator of the PI services. As such, the Shipping Service is responsible for receiving PI enabled orders and with the usage of the capabilities offered by other services, make appropriate decisions to ensure the delivery or handle the non-delivery of an order in an end-to-end manner.

Through it's various integrations with the rest of the ICONET components, the Shipping Service will have all the data needed to make appropriate decisions during the order's lifecycle. The integration with the IoT Cloud Platform (D2.8) enables it to have information about sensor values (such as location, temperature etc.) to provide the possibility for informed decisions based on the order's requirements. Additionally, the integration with the Web Logistics service, and its blockchain implementation (and Smart Contracts) allows for validated checks for violations of the aforementioned requirements. The key inputs and outputs of the shipping service are be outlined in Figure 5 below.

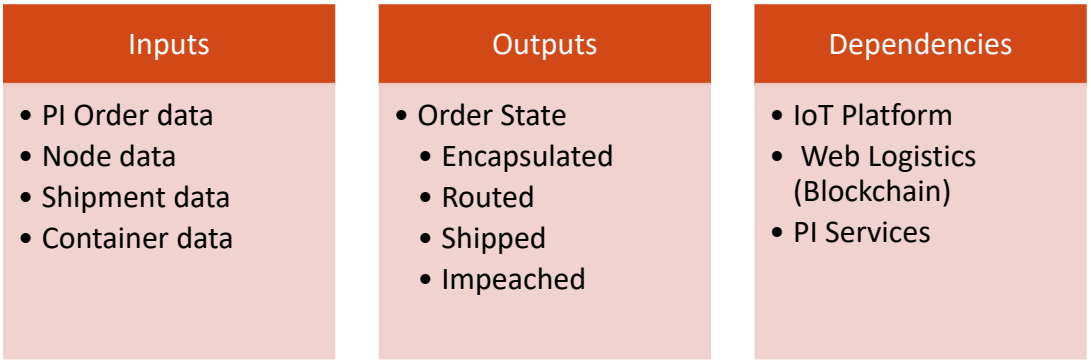

Figure 5 Inputs, Outputs & Dependencies of Shipping service

The decisions that the Shipping service is responsible for, as mentioned previously, is best described as a state diagram, shown in Figure 6. below.

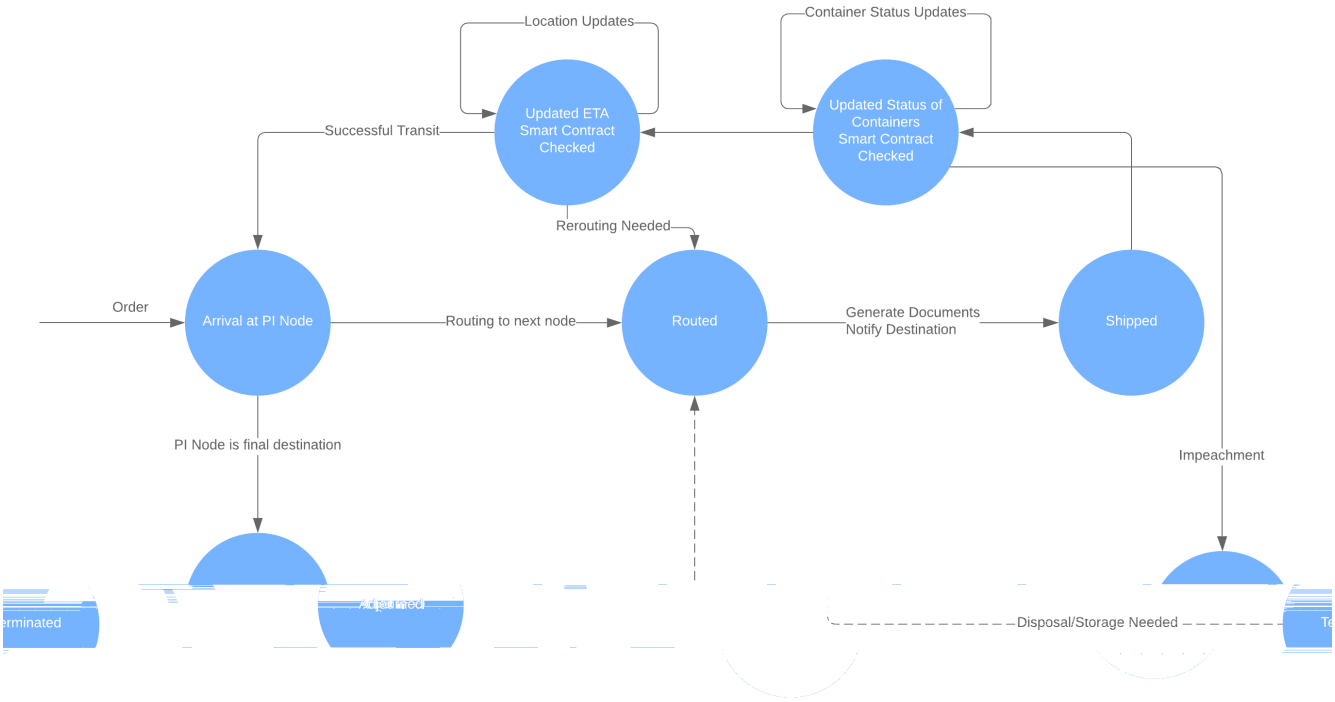

Figure 6 Example of Shipping state management

Figure 6 shows an order arriving at a PI node, after which the Shipping is responsible for making all appropriate decisions to forward it to the next destination (more analysis on the specific interactions between services will

be presented in section 5 with a full example of an order transversal). After this is done, the Shipping Service will continuously check the data received from the IoT sensors residing in the containers used to transport the order and check the values against the requirements established in the Smart Contracts. In parallel, appropriate status updates are made to the state of the order. In case of an impeachment, the Shipping service is responsible for terminating the shipment and if needed, rerouting it to a valid disposal location. If the order progresses nominally, these actions are repeated on every PI node the order passes through (if necessary).

4.3.2 Encapsulation Service

The Encapsulation Service, as mentioned earlier, is based on the Encapsulation layer of the OLI & NOLI models. For the purposes of ICONET, while no technical implementation will occur, sometimes actions that would probably fit more into the description of the Physical layer or a potential Physical Service are also included. This occurs on a case by case basis and is not considered in scope for the Encapsulation Service. The role of Encapsulation in ICONET is described as the process of "encapsulating products in PI containers". In the duration of the project, we have concluded that the Encapsulation service needs to also encapsulate PI containers into larger sets of PI containers and packing these PI containers into PI Movers (trucks, trains, etc.). Since the algorithmic problem described in these functions has the same basis, i.e. Three-dimensional bin packing problem, a unified solution provided by the Encapsulation Service can satisfy all of these requirements.

It can be useful to think of these processes as a multi-stage encapsulation, with the 1st stage being the encapsulation of products into PI containers, the 2nd stage being the encapsulation of PI containers into larger PI containers etc. These different stages of the encapsulation process, while having the same basis, have different constraints, which the Encapsulation service will address. The 1st stage encapsulation is done based on the dimensions and type of the products and containers, meaning a product has to fit into a container to be encapsulated and if the product needs refrigeration for example, the container has to be able to provide that. For the 3rd stage Encapsulation, (PI containers into PI movers), other considerations can apply, such as weight distribution of the vehicle, air between products/containers, as well as cargo consolidation based on routing decisions to minimize empty space, etc.

The encapsulation services' key inputs, outputs, and dependencies are presented in Figure 7 below.

Figure 7 Inputs, Outputs & Dependencies of Encapsulation service

4.3.3 Routing Service

As described in D2.3, the routing operation for the PI is "The Routing operation selects the feasible/optimal routes (out of those identified by the Networking Layer) through the PI that connect the origin of the shipment (i.e. the initial π-hub handling the π-units comprising the shipment to the final destination/ π-hub that will handle the shipment". Put simply the routing services finds the best possible route through the network provided by the networking service. There are additional constraints that are also considered in route calculation including but not limited to:

Reducing number of empty transports
Travel Time
Cost
Consolidation of Containers
Perform both Delivery and Pick Up

It should be noted that these constraints are not static and will vary according to the specific use case addressed. For example, LL3 e-commerce has a heavy focus on optimising routes that include both delivery and pickup operations using multiple trucks, where LL1 PI-Corridor focuses on end-to-end travel time and on time deliveries. The routing algorithm developed with the broadest application was VRPM (Vehicle Routing Problem with Backhaul). The inputs needed for this algorithm are:

List of Pick Up Nodes
List of Delivery Nodes
List of Vehicles
Supply Information per Pick Up Point
Demand Information per Pick up Point
Start & End Times for Delivery Service
Service times per Node
Travel Costs
Travel Times
Max Vehicle Mileage
Vehicle Capacity

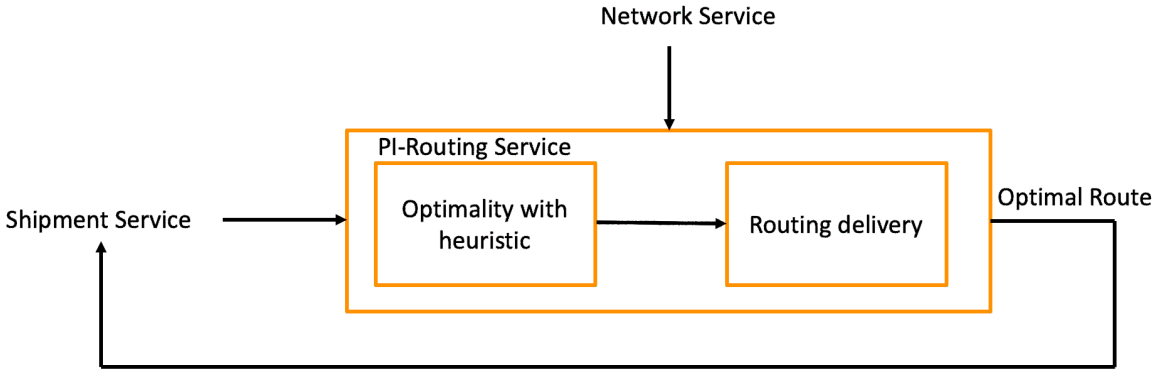

Figure 8 Routing Service Integration

The source for much of the inputs is the Networking Service (which can provide info about the layout of nodes (distance, travel time etc.) and the nodes themselves (service time, supply & demand etc.). The remaining information is supplied by the logistics operator (vehicle costs, vehicle capacity, start & end times of delivery service etc.) and this will be forwarded through the shipping service which manages the overall order. *Figure 8 Routing Service Integration* shows the interrelationship between the routing, shipping and networking services.

The output from the service is an ordered list of nodes representing the optimal route (usually by cost but not always) as well as an estimated cost for the service. The service can be re-utilised after each node is reached, thereby continually ensuring the optimal route is taken by accounting for real time updates coming from the vehicles, the nodes or external factors. Again, this information comes via the network service or the shipping service and can utilize analytics coming from IoT tracking systems that the shipping service consumes.

4.3.4 Networking Service

The Networking Service is responsible for collecting, storing and integrating all PI network information and communicating PI shipment relevant network information in an interoperable and interconnected manner, as captured by the Network Layer. It is therefore, divided into two separate stages, Stage 1 dealing with the collection and integration of PI network information, and Stage 2 dealing with the provision of PI Shipment specific information. Starting from identifying the relevant nodes and links of the network(s) in collaboration with the Link and Physical Layers, the Networking service Stage 1 classifies them in terms of geographical location (and scale), transport mode, and level of aggregation. For this purpose, the GPICS PI Node and PI Link typology is adopted. In Network Service Stage 2/ Step 1, this classification is utilized through the application of the Area, Modes, Aggregation level and Data detail tools, which are described as follows:

The aim of the geographical scale function (Area tool) is to limit the scope of the search area for network components. An area of relevance is identified on the global map, for the specific PI shipment submitted. For example, the scale will be different for a request to carry a cargo from North to South Europe, or between two neighboring French cities. The Area tool utilizes the origin and destination coordinates of the PI shipment to identify an oval shaped area of relevant PI network components.

The mode tool considers restrictions imposed by each PI order on the modes available for shipment. If more than one mode is feasible, the Networking Service assess the multimodality options available, by considering transshipment nodes and various mode links.

The aggregation tool considers the level of detail required for the routing request associated to a specific PI order. PI nodes can represent international or local hubs in the case of long-haul shipments, local warehouses and postcodes in the case of e-commerce, as well as specific function (e.g. customs) in complex port (intra-hub) operations.

With awareness on the scale, aggregation level and modes a final decision is made on the level of data detail required. This will depend on the PI order made, but also on data availability. The output data detail can range from physical properties of infrastructure, to live information on the services operating on the PI network. Four levels of data details can be identified:

Infrastructure properties: Network information describe static infrastructure characteristics such as the length of a link, the modes that can accommodate the function of carrying cargo (e.g. truck, rail), or even more detailed information such as classification into motorway, or number of lanes. A similar

concept can be applied to the description of nodes. A node can represent a warehouse that has specific capacity for storage and docking capability.

Operational status: Additional to network static information, operational condition information is collected for each PI network component. For example, incapacitated links due to roadworks or traffic, will have their travel time and travel cost (if applicable) updated with most current information. Similarly, for nodes up to date availability and status of infrastructure is considered, such as spare capacity.

Services schedule: The aim of this data layer, is to account for the fact that roads and warehouses do not handle directly cargo transport, but rather indirectly through services. Truck, air, river/sea or rail transport hauling services that utilize the infrastructure are responsible for undertaking the task of physically moving cargo from one location to another. With that in mind, at Stage 2/ Step 1 the Networking Service collects (if available) service schedules that operate between specific locations.

Services status: A final layer of complexity can be anticipated, that accounts for live information on the capacity of en-route hauling services and queues at PI hub services.

It is also essential to associate each physical infrastructure component with properties to be used for operational decisions. Networking Service Stage 1/ Step 2, deals with this requirement through four tools that adjust the information collected in Step 1 to specific PI Order requirements.

Link filter and node filter tools capture the restrictions imposed by the PI Order and filter out any network components that do not meet them. For example, for a shipment that requires refrigeration, the respective PI node and PI service property is examined, and nodes and services that do not offer this option are not considered further.

The Link and Node Cost tools calculate the final metrics for each PI component in terms of KPIs as requested in the PI order, such as travel time, travel cost, service reliability. Performance is adjusted for each PI component considering the requirements, constraints and limitations implemented in the previous steps of the Networking Service.

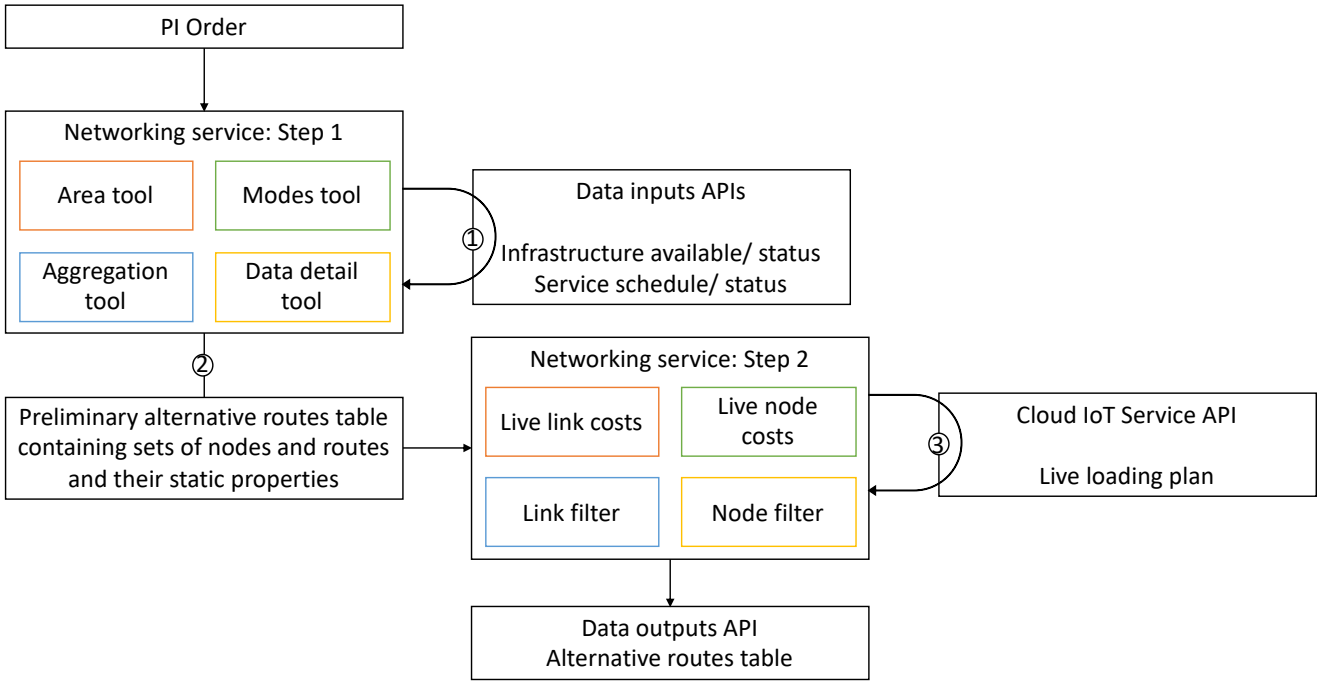

Figure 9 Interactions of Networking Service

The final output of the networking service is a set of interconnected PI nodes, PI links and PI services, that are available for transporting a shipment between any two nodes. Stage 1 of the Networking Service that is responsible for collecting information on the status of the network, is always listening for changes in traffic or services status. Depending on the nature of the PI Order the Networking Service Stage 2 can be either called once or several times. For cases that only static information are available the Networking Service is called only when an order is initiated, while for cases that information is dynamically updated, it is called whenever a shipment arrives at a PI node that it is not the destination. Figure 9 captures the operational sequence for Stage 2 of the Networking Service.

5 Traversing a PI-network

This section attempts to identify and map the processes that a PI order must go through while traversing a PI-network, identify the events that are key to initiating these processes, as well as understand the decisions that need to be made at every step. Additionally, the data needed at every step will also be identified. To achieve that, a scenario of a shipment travelling through the network will be described. While the previous version of this deliverable also documented this scenario, it has been updated to include a more complete image of the decisions and operations occurring in the shipment's trip, as well as attributing the decisions and functions on every step to specific services, while highlighting their interaction points.

5.1 Scenario Parameters

The selected scenario describes a pharmaceutical cold-chain, under which, the storage, handling and transportation of the shipment is sensitive to external conditions and requires certain temperature and humidity, to avoid compromising the chemical stability of the product. Cold-chains might also require specific conditions in terms of exposure to light, vibrations during handling, and shocks.

For this scenario, the manufacturer (or the regulator) must define the constraints for all supply chain processes. In more detail, we assume an acceptable temperature range between 3° C - 6° C. The acceptable humidity range is 30% to 40%. The quality of service requirements include that the product must reach its final destination within 5 days after dispatch. The manufacturer also proposes that the shipment, should not be stored close to doors or windows to avoid compromising the integrity of the product.

5.2 Scenario Goals

The objectives of this scenario are the following:

1. The product must arrive within the defined timeframe (<5 days)
2. The product arrives without being compromised (temperature 3° C - 6° C and humidity 30% to 40%.)
3. Cost is minimized
4. Environmental impact is minimized
5. All participants are accounted for and reimbursed for their services

5.3 Key Steps

5.3.1 Consolidate Order information & constraints

The scenario begins with a shipment of the cold-chain pharmaceutical product, that needs to be sent from Point A to Point B, while maintaining the criteria defined in the previous section. The first step is to propagate the information of the shipment to the PI network for handling and shipment, in tandem with necessary physical actions, such as the shipment being physically in the starting point. For the PI aspect, the Shipping service will register the order as communicated by an external system. This in turn will convert all relevant information to the PI data model which will be used across all services of the PI. This new PI order will also contain all the constraints (described above) in order to enable the services to make optimal decisions. These constraints are stored in the Web Logistics service using blockchain enabled SLAs which are used for verification and live checks of sensor values indicating whether the shipment operates within acceptable values of the constraints. The first

decision that needs to be addressed is how this shipment will be packaged and encapsulated, while minimizing cost and environmental impact by reducing empty space in cargo transports, but more importantly making sure that the targeted quality of service is adhered to. Assuming that the dimensions of the shipped products are known, the encapsulation service is responsible for optimally packing the products (either directly or their packaging) into PI containers. These containers are a core part of the shipment's travel through the PI network, as the sensors embedded in them are responsible for propagating all the responsible information to decision makers in the PI as well as interested parties(location, temperature etc.).In this case, a container that is refrigerated with temperature and humidity sensors would be chosen.

After the products of the shipment are encapsulated, the order status is updated and is marked ready for routing.

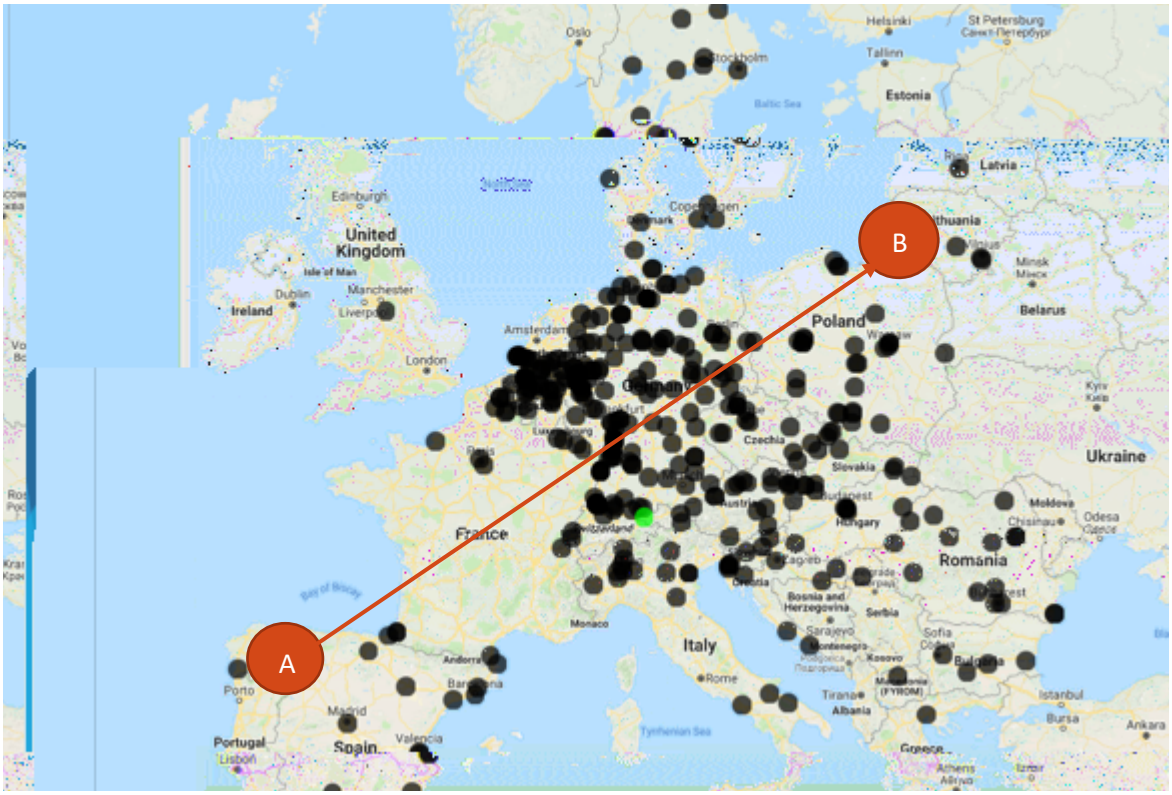

Figure 10 Shipment requirements

For this specific example, this means that the shipment must reach Point B, within a certain timeframe (5 days) and in a certain condition (not compromised, spoiled or damaged), as seen in Figure 10. The details of the shipping process, the requirements and constraints as well as the financial aspects of the shipping process have already been defined by the shipping service, as described in more detail D2.3 and D2.4. What is left is to define the best route through the PI-network for point B. The routing service needs to be aware of all possible paths from Point A to Point B (Figure 1111).

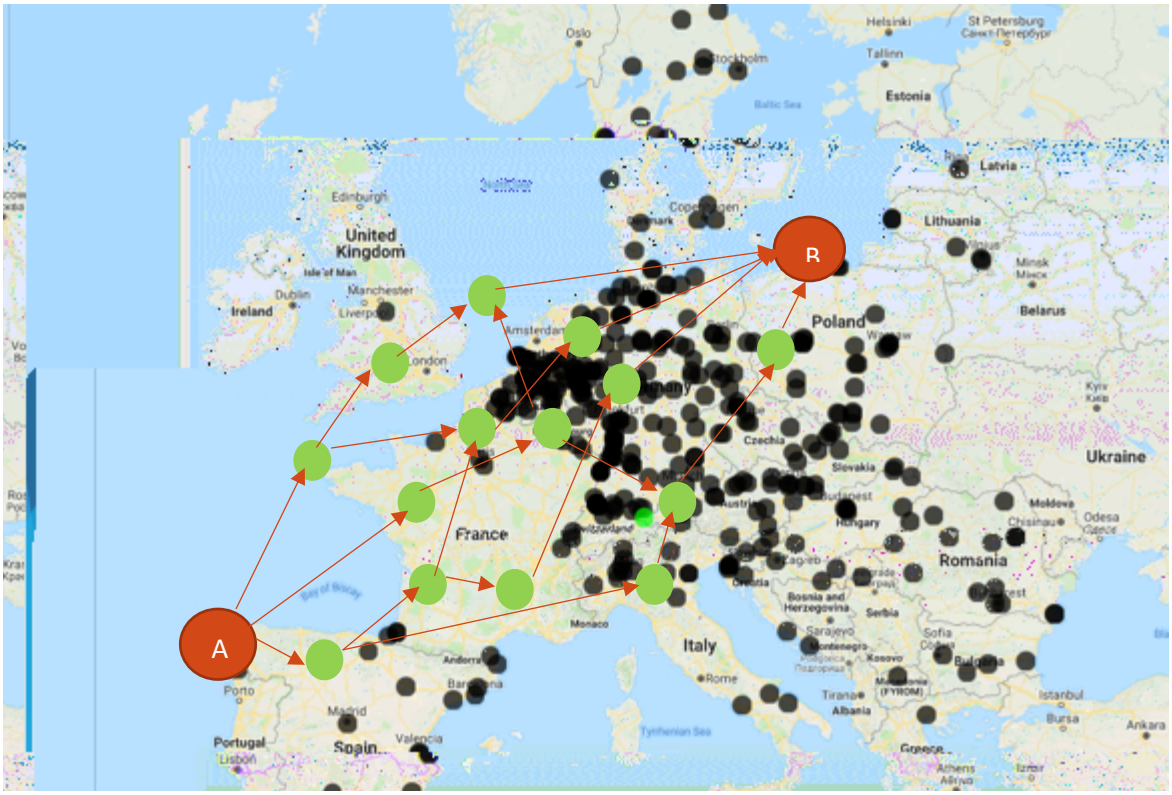

Figure 11 Identify all paths

The Networking service contains information regarding PI node capabilities and the routes that connect them. In this instance, the networking service would take into consideration any constraints of the order to return only paths that are in line with these requirements. That means that the paths returned will have an ETA < 5 days, and if storage is needed in a node through the trip the node selected at any point should support be refrigerated and support the required temperature 3 – 6 degrees Celsius and humidity of 30% to 40%. As such, all possible paths that are compatible with the shipment instructions and constraints, as defined earlier, are passed on to the Routing service from the Networking service. Having identified all possible routes, the routing service then 'decides' on the best route (at the time) (Figure 122), taking into account the instructions and constraints, as defined by the shipping service. The routing service will also need to be aware of the conditions at each PI-hub, taking into account capacity, utilization and congestion between hubs.

In contrast to the digital internet, where the number of hops is a key metric, the physical distance between hubs is important as it affects cost, environmental performance and quality of service. The routing service will be 'called' at every hub, when a shipment arrives, in order to ensure that the planned path is still the 'optimal' and update it if necessary. Therefore, the arrival of a shipment at a hub is the event that triggers the routing service, taking into account updated information regarding the conditions at the next hubs. The routing service also requires information regarding all possible paths forward, as defined by the networking service. This means that 'calling' the routing service will also trigger the networking service for an updated of possible routes. Changes in the predetermined route based on these routing iterations, could also potentially trigger the Encapsulation service. Further bundling or unbundling of cargo can be done through this encapsulation iteration, always adhering to the constraints set up initially. These iterations allow for potentially much greater savings in terms of volume used and CO2 emissions, as the various PI movers can be loaded/unloaded on hubs located between Point A & Point B.

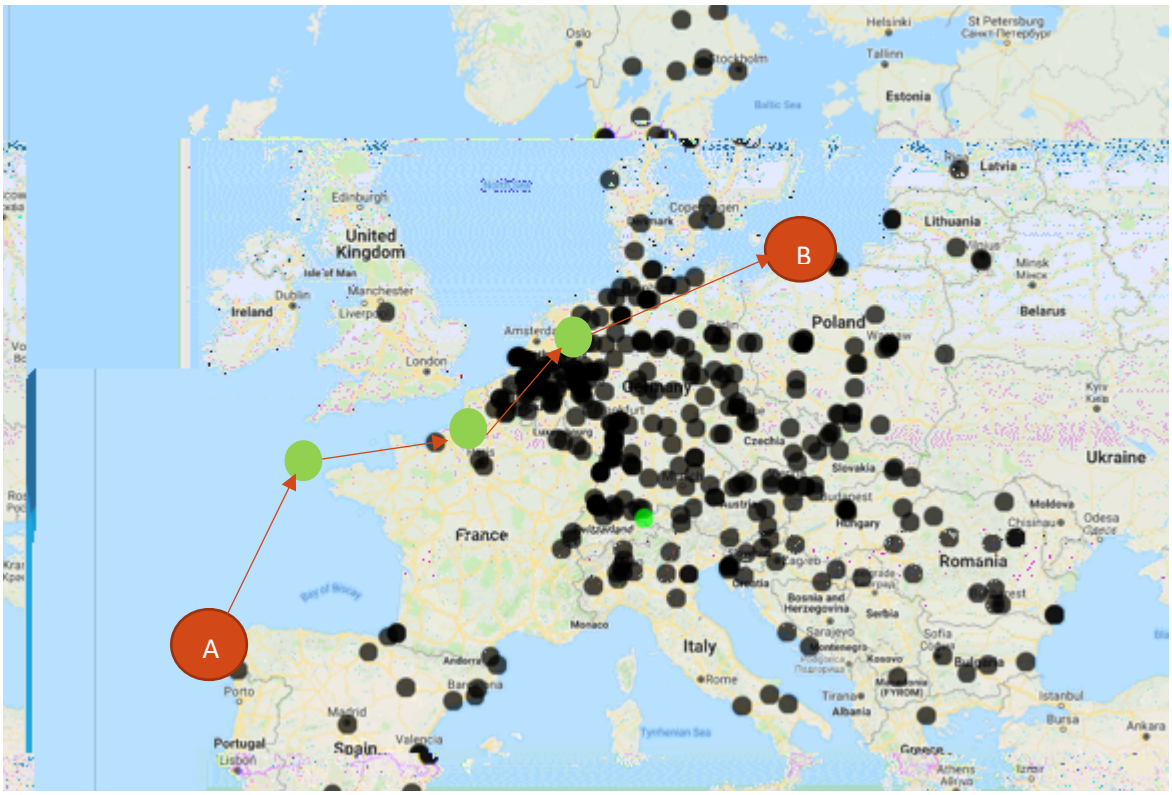

Figure 12 Selected path

5.3.2 Transportation

Once the best path has been identified and the preferred transportation have been selected, the PI-container begins its journey in the PI-network. The selected means of transport must also adhere to the quality of service requirements, ensuring the integrity of the shipment as well as the timely arrival of the product. For this scenario, the selected transportation must offer a controlled environment (in terms of temperature and humidity) as well as monitoring capabilities. Monitoring capabilities must cover coordinates, temperature, humidity and status of the shipment. The margin of error for the key metrics must also be provided.

5.3.3 Arriving at a hub

When the shipment arrives at a PI-hub, some of the previously made decisions must be revaluated. Arrival at a PI-hub is an event that will trigger revaluation of the selected path and transportation means. This means that the routing and networking services will have to be 'called' again to evaluate potential new hubs and paths, identify better paths to the final destination and the required means to get there. In cases where the initial path is modified, the shipping service will have to be called in order to update the shipping agreement accordingly.

If the shipment has been identified as compromised, a new route will have to be defined. The nature of this route will depend on the instructions of the shipment. This might be a new reverse logistics network to return the shipment to the sender or a new route to the appropriate disposal of the shipment. In both cases, a new order may or may not be initiated. To achieve that, the routing, networking and shipping services will have to be 'called', to identify and evaluate possible paths. The shipping service will define the actions to be taken in case of a compromised shipment, while updating the shipping agreement accordingly.

In cases, where the shipment needs to be decomposed in order to reach different final destinations, the encapsulation service needs to be 'called'. In the case where shipments are split and combined, the encapsulation service will need to be aware of specific requirements, as defined in the shipping agreement. Relevant information includes the type of the product, potential issues with cross-contamination, physical information such as product form and more importantly handling instructions. For the case of the cold-chain pharmaceutical product of this scenario the encapsulation service must take into account the controlled environment requirements and handling instructions. For example, this shipment should not be composed into larger shipments containing products with no specific temperature requirements. In contrast, if a shipment was to be composed into a larger freight, the encapsulation service could perform a second encapsulation of containers within containers in order to reduce empty space and by consequence, transport costs. From a practical perspective, using a cold-chain to ship products with no such requirements, will not only increase the cost of transport but might also affect all products.

5.3.4 Arriving at the final destination

This scenario is concluded when the shipment arrives at the final destination. Once more, the final destination must have the capability to evaluate the condition of the shipment, ensuring that temperature, humidity, and other handling requirements have not been compromised. If the shipment has been identified as compromised, a new route will have to be defined as described in the previous section. In cases where the shipment fulfils the given conditions, the shipping service will have to be called in order to update the shipping agreement and create a 'goods received' note to be sent to the sender. The shipping service will also need to ensure that all network participants are notified, their services are accounted for that the relevant receipts are produced.

5.4 Conclusions and impacts to the architecture

The updated defined scenario illustrates a high-level case study of a PI-network, while showcasing the requirements, key and interactions made by the services. These requirements include technical aspects, decisions, events, and information/data flows that are needed in order to realize the vision of PI, and as such have heavily influenced the reference architecture.

5.4.1 Technical requirements

The technical requirements, identified in this scenario, include the need for monitoring across all steps in the PI-network. Monitoring should cover all the quality of service, handling, storage and transportation requirements. For this scenario, these include monitoring of temperature, humidity and integrity of the shipment. IoT sensors must also capture the location of the PI-containers throughout its journey. These requirements are addressed with the use of IoT sensor technology, as described in D1.6.

5.4.2 Decisions

The described scenario also addressed the decisions that need to be considered at each step. The key decisions that have been identified are the routing, transportation and encapsulation decisions that need to be re-evaluated at every step, to ensure that the shipment is at the best path to its final destination, taking into account updated information regarding capacities, congestion and other operational factors. All decisions will need to adhere to the requirements set by the sender in terms of quality of service and other shipment constraints.

5.4.3 Events

The identified events that occur in the initialization of a PI order are show cased as a flow diagram in Figure 13 below:

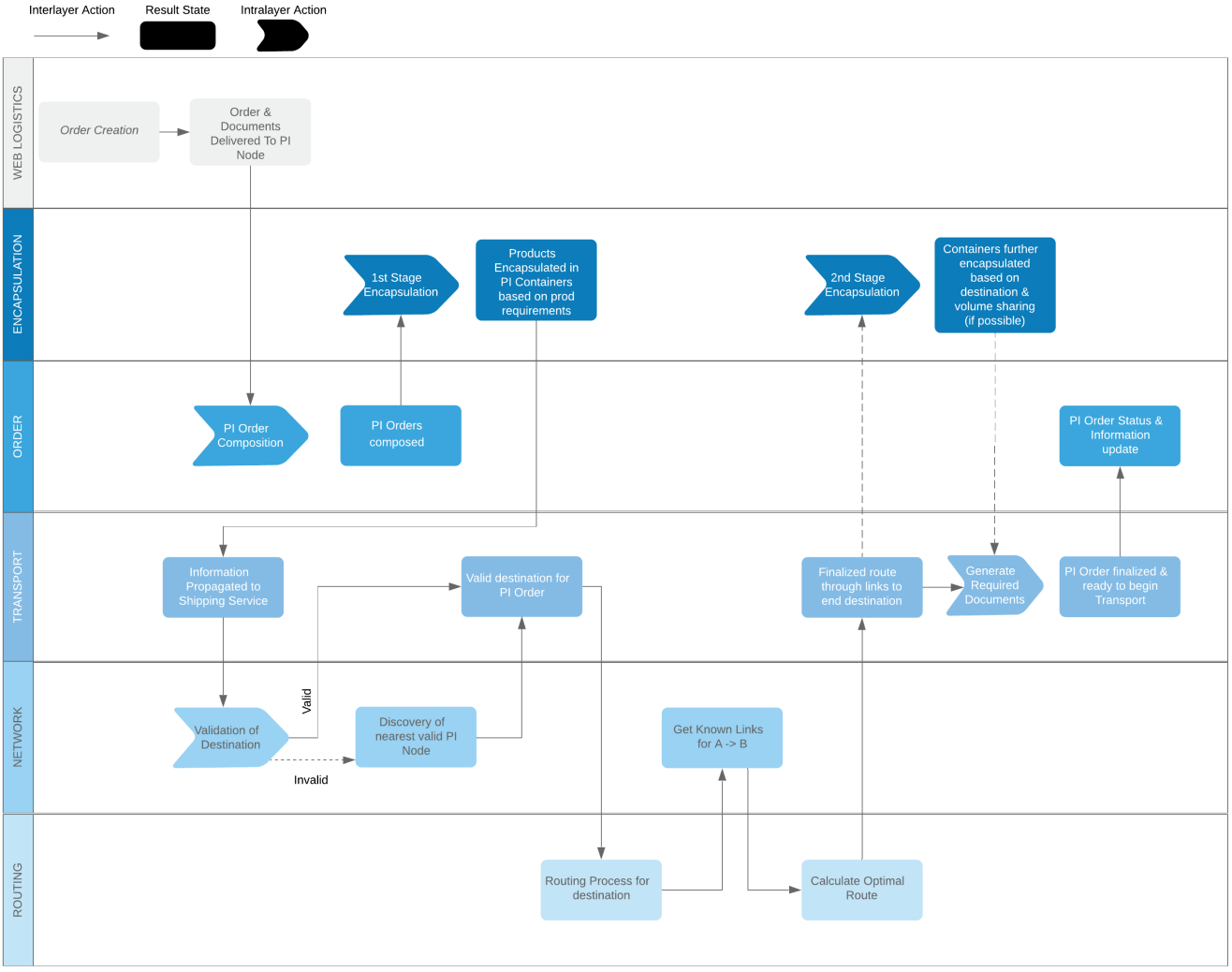

Figure 13 Initial events of shipment

These events are:

Order: placing an order for a product will trigger the planning of the shipping process, creating SLAs and composing a PI order with relevant constraints

Encapsulation: products of order are packed optimally based on product requirements

Networking: validate destination is in line with order constraints

Routing: optimal route must be identified

Shipment ready for dispatch: when the shipment is ready to be dispatched, the shipping agreement must be generated and the order is shipped

Shipment arrives at hub: when a shipment arrives at a PI-hub, routing decisions and in some cases encapsulation decisions should be re-evaluated ensuring that the best path is followed

Shipment arrives at final destination: when the shipment arrives at the final destination, a receipt must be generated, and all participants must be reimbursed for their services

5.4.4 Data requirements

As a first step towards the conceptual PI architecture, the previous version identified the high-level data specifications for different services and systems within the ICONET universe. The physical layer , while not being used as a service by ICONET, has been defined as the network of all physical elements of a supply chain network. As such, the elements of the Physical Layer were identified and described. The results of this identification process was used to guide the initial formation of the ICONET ontology. This version was iterated upon during the project's previous months, based on the needs of the services and systems that were being developed. Furthermore, special care was taken in order to shape the next versions of the ontology in a manner that allowed for the seamless integration of the ICONET services. These definitions allowed for the creation of the conceptual data structures and flows, that were later validated through the simulation service and the living lab use cases ensuring interoperability through different scenarios and scales of PI implementations. The

The previous version of the report identified the high-level requirements for key entities of a PI network. This included baseline container data of the PI container shown below (Figure 14).

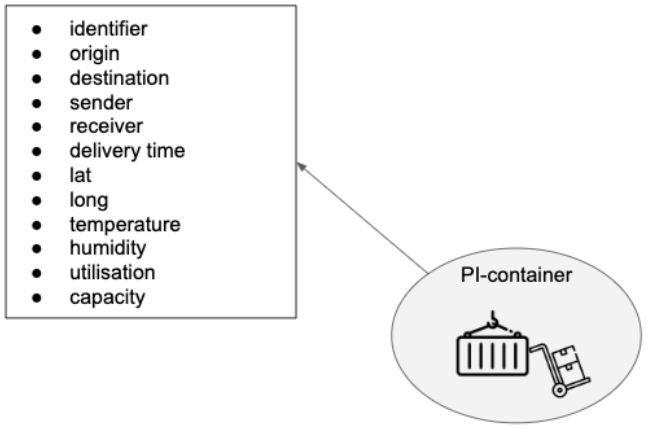

Figure 14 PI-container data specifications

A PI-container must be uniquely identified, hence the need for an identifier. The identifier could be linked to the IoT device that is monitoring the container and it should be linked to the various identifiers the container might have in other systems (e.g. order id, shipment slip etc.) to facilitate provenance and traceability. Monitoring the status and location of containers also raises the need for operational information such as the location (latitude, longitude), delivery time, capacity, utilisation origin and destination as well as product dependent information such as temperature and humidity. To facilitate the operations of the IoT sensors as presented in D2.7 , a more concrete data model was created. Two same level data structures, Shipment data and Requirement data were used containing the information needed. The shipment section contains the identification data needed for linking PI containers with shipments and the requirement section contains all configuration data for the various sensors of the container, allowing the transmission of alerts if values detected outside of those described in this section are not within acceptable ranges.

A similar approach has been adopted for PI-nodes, that receive PI-containers. The relevant data for PI-nodes is presented in Figure 15.

Figure 15 PI-node data specifications

All PI-nodes must also be uniquely identifiable. It is also important to be aware of the level of a node (as defined in D1.7 – L1, L2, L3 for Country, NUTS-2 and Urban level nodes). In the process of identifying the optimal route for a container, the networking

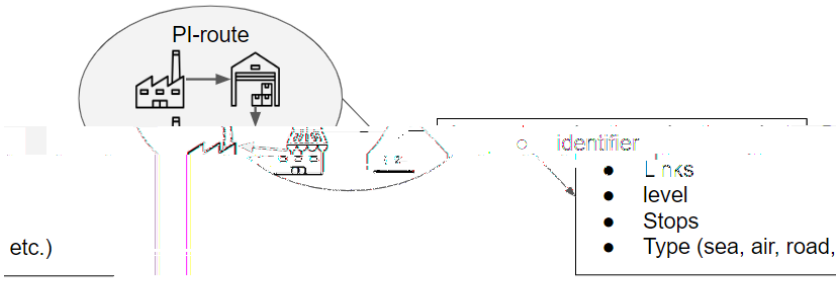

Figure 17 PI-route data specifications

Link data will be extracted from existing sources, such as TEN-T corridors in a Pan European level, or shipping route information for lower levels. Links are also a key component for which data is required in realizing the vision of ICONET. The data is presented in Figure 18.

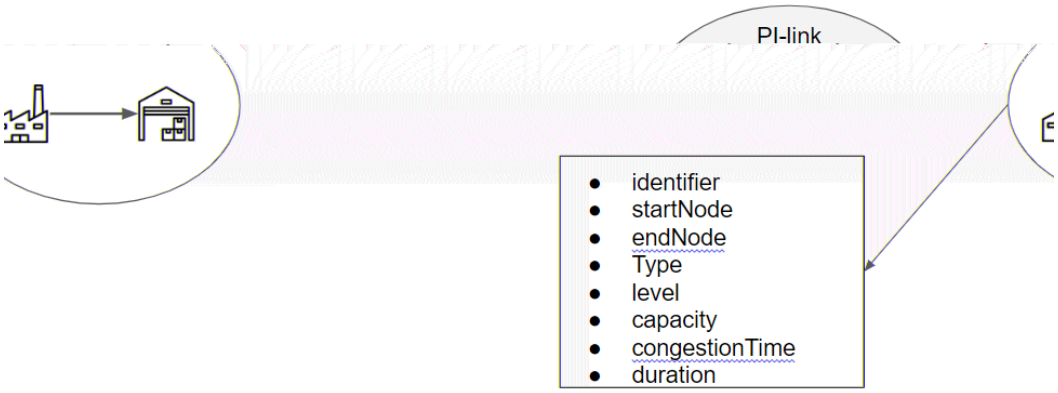

Figure 18 PI-corridor data specifications

Sets of links form routes and the information that is needed to evaluate and optimize the selection of a link, include the nodes of the corridor, their type, the capacity of the corridor, duration of the trip as well as information about any congestion that might occur, as well as the level they operate on. Given the dependencies presented in this section, the PI-network is defined as presented in Figure 19.

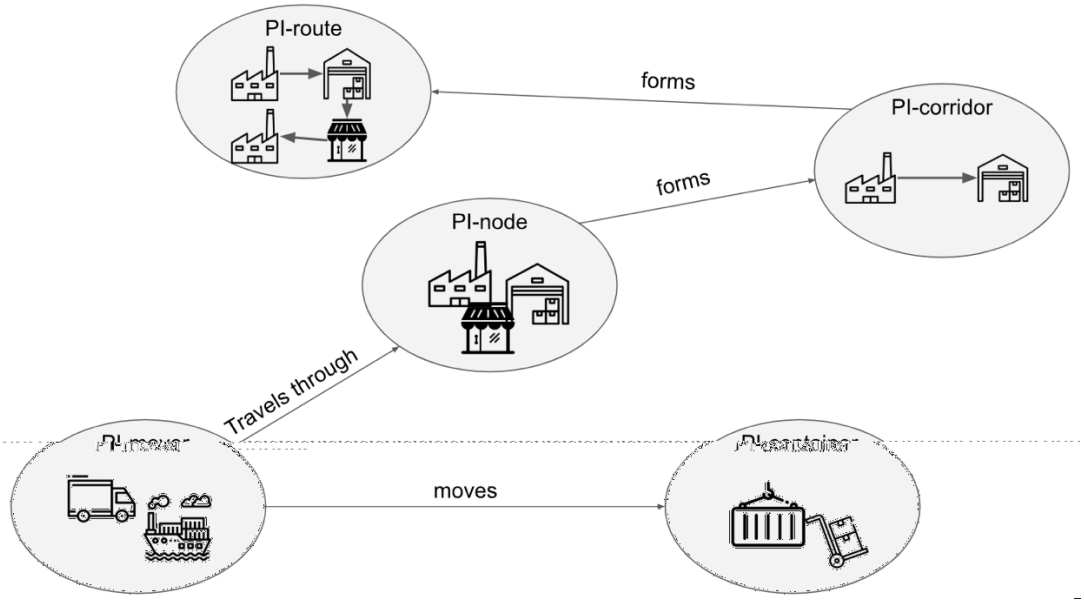

Figure 19 PI-network dependencies

5.4.5 Integration with existing/legacy systems using PI Data Adapters

The physical layer represents the physical elements of a network, responsible for generating data. The data layer represents the tools and methods responsible for capturing the data. It consists of two components, IoT component, accounting for all the IoT devices that are responsible for the connectivity and the extraction of data and the legacy components, accounting for the legacy systems that already exist across the supply chain partners.

At the legacy component, it is important to Identify all relevant legacy systems and investigate integration options. Integration can be achieved through APIs or custom adapters developed within the ICONET project in a limited scope. Legacy systems are another source of data that will feed into the ICONET services to realize the vision of PI.

The complexity of manufacturing operations and that of logistic operations, have led to the development of a cornucopia of enterprise systems. From MRP (Materials Resource Planning) and MRP-II (Manufacturing Resources Planning) to ERP (Enterprise Resource Planning), enterprise systems are still growing in scope and quantity, integrating almost all downstream and upstream operations, aiming to offer an end-to-end supply chain system. For the purposes of ICONET, enterprise systems are classified as internal, external and extensional.

External systems are the systems not directly related to logistics operations but offer significant information sources. These include systems that contain information regarding traffic, general search services and other related sources. More specifically, ICONET external systems might include services like Marine Traffic, offering real-time information on global ship traffic, weather data sources, offering weather information that can be input to various services, routing data (e.g. Google maps) offering information about available modes of transport, start and end dates, load data, and waypoints. Routing services can also offer traffic data, an invaluable input to almost all optimisation services.

Extensional systems are defined as systems that extend ICONET functionality. They are also external systems but they are not information sources but rather services such as optimisation solvers.

Internal systems are defined as the systems used in logistics operations. These are the legacy systems used by ICONET LL partners. They include numerous enterprise systems such as ERP systems, Warehouse Management Systems (WMS), Transportation Management Systems (TMS), Manufacturing Execution Systems (MES), Inventory Optimisation Systems, Procurement Systems, Production Planning Systems, Scheduling, Transportation Planning, Order Management Systems and Demand Forecasting tools. Input and output data for these systems must be identified and documented, connectors/adapters must be developed where necessary. These data must be supported by the ICONET ontology.

As a first step towards the conceptual PI architecture, it is key to identify the data requirements for different services and systems within the PI universe. Identifying data requirements will inform the ICONET ontology and the initial architecture as they will drive the integration-related decisions as well as the service modelling. The first step is to identify the data available at the legacy system level. Mapping available data will enable the definition of data structures, definition of data flows and will eventually facilitate the development of ICONET ontology, that will enable system interoperability. The following data was considered in the course of the project and the ICONET ontology was influenced and included a number of these data structures.

Internal systems data include data from ERP, WMS, TMS, MES, IMS, Order Management and other planning systems. For ERP systems, the following data have been identified:

- User details, such as ID, Name, Phone, Title
- Account details, such as Name, ID, Type
- Contact details
- Campaign details, such as type and status, in order to track success rate or any other key performance indicator (KPI)
- Reporting
- Lead time
- Mapping information (to avoid syntactic issues)
- Advanced shipping notification

For WMS, the following data have been identified:

- Master data: Article number, Description, Article weight, Article length, Article width, Article height, Quantity unit, Type unit load, loading factor, gripping unit, Blocking indicator, Batch number, Weight/retrieval unit, Weight/unit load, Client, Best before date, Remaining runtime, Sorter capability
- Inventory data: No. articles, Total stock, Average stock, Minimum stock/art., No.UL/art., Available stock, Shortages
- Movement data: Goods receipts/ day, Goods issues/ day, Storages/d, Retrievals/d, Quantity transship./a, Restorages/d, Orders/d, Orders per article, Positions/order, Positions/d, Grips/ Pos., Incoming orders/h, Order lead time, Material lead time, No. of orders/ order type, Double cycles/d, Complete units/d

- Other systems data: Order types, Unit load master data, Packaging master data, Storage capacity, Space restrictions, Room restrictions, Utilization space/volume, No. UL/art., No. staff/dept., Operating costs (manpower, energy, maintenance), Investment costs (replacement), Value turnover/a, Productivity

For TMS, the following data have been identified:

- Modes of transport available: capacity, cost, fuel consumption, load restrictions, environmental footprint
- Schedule data: intermediate stops
- Product data: size, weight, volume, special conditions, best before date
- Routing data: routing information, high, low and average speed per mode of transport, transportation costs, paths, connectivity groups (intermodal)

For MES, the following data have been identified:

- Lots or batches coming in or being created
- Lot and batch attributes describing material
- Operating supplies used
- Machines used
- Process data obtained
- Individuals involved in the production process
- Tools used
- Repairs of machines and tools
- Quality data (such as measured values, inspection equipment used, inspection
- decisions)

For IMS, the following data have been identified:

- Items
- Item number
- Item Description
- Item categories
- Item cost data
- Item price data
- Inventory movements
- Location
- Available quantities
- Item suppliers
- Item manufacturers
- Item customers
- Lead time
- Best before date

For production planning systems, the following data have been identified:

- Current status
- Start, end and latest update dates
- Quantity
- Criticality
- Owner
- Data source

For demand planning systems, the following data have been identified:

- Name, description, and relevant categories of the sale
- Spatial and temporal data, such as Location and Due date
- Priority
- Status
- Minimum shipment and maximum lateness dates
- The Item related to the sale
- Quantity of the Item
- Owner of the Item
- Category and price of the item
- The Item's source
- Materials required
- Job details
- Projected materials requirements
- Historical data on material requirements
- Customer service levels
- Reorder point
- Economic order quantity
- Suppliers data

Various sources of external information have also been identified. This use of external data has two objectives. The first is to act as a complementary source of information, offering data that is not available in ICONET systems, while the second is to offer data that can be used for validation of data obtained from ICONET systems. For example, data obtained by IoT devices monitoring the location of PI-container can be cross-checked using global ship traffic data. The identified relevant sources are listed below. This list is by no means exhaustive.

Another type of information that might affect decision making in logistics operations is weather related data. Adverse weather conditions might affect the estimated time of arrival for ships or even trucks. The following types of information have been identified:

- Current weather conditions at a location
- Forecasted conditions at a location

Routing data are expected to be key for the realisation of PI. They are also expected to be invaluable input for many of ICONET's offered services. The following routing data have been identified:

- All possible paths
- Distance of paths
- Time of paths
- Coordinates of paths
- Instructions

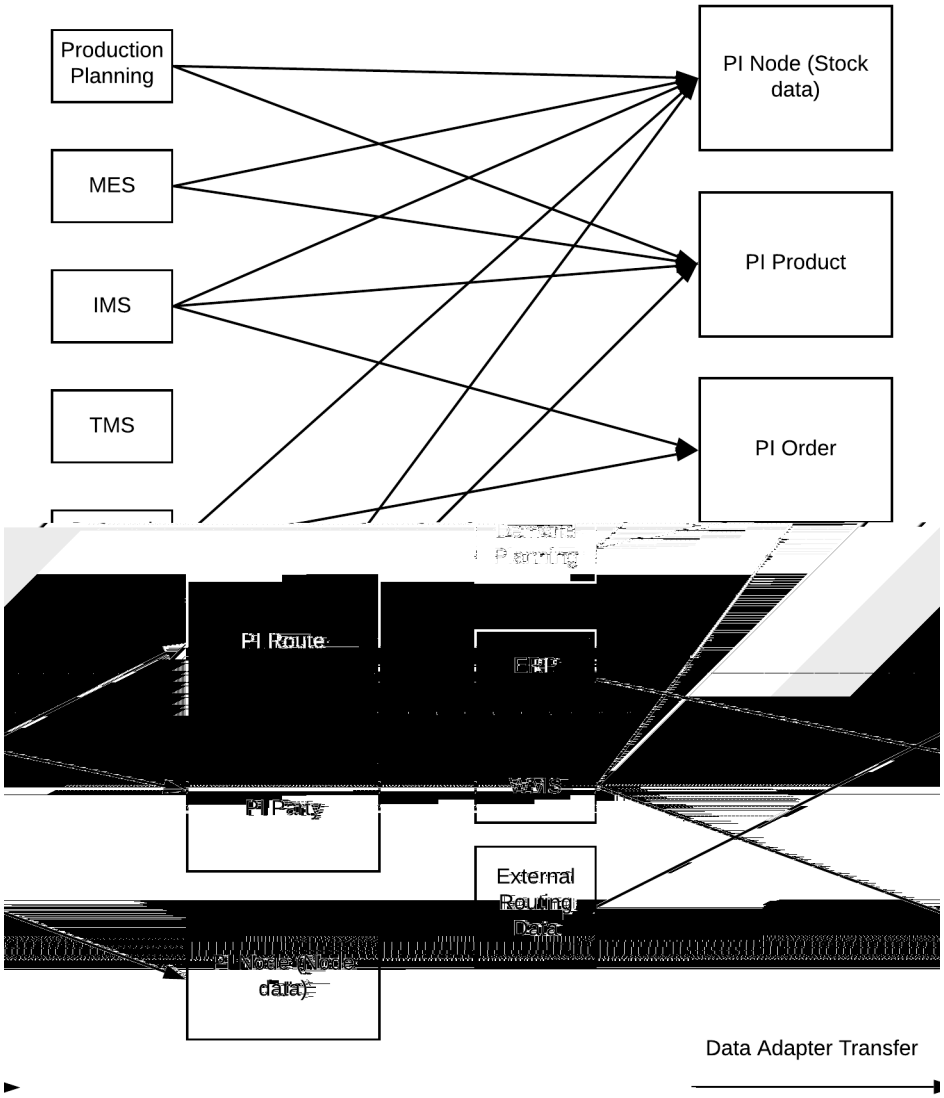

Figure 20 External Data Integration

These Data Adapters will also have to address issues regarding confidentiality, protection and security of data. While the technical implementation is not in the scope of ICONET for these adapters, from an architectural perspective, multiple solutions can be proposed. As is expected, these solutions might not apply to every single case since the adapters will need to be built on a per use basis. Such solutions could include:

- Air gapped data adapters for local system installations. Disallowing any network access means that the data is not accessible from outside and thus is safe from potential malicious third parties
- Secure communications protocols, such as HTTPS or Oauth2 based protocols should be used for Data Adapters communicating through APIs
- Anonymization protocols, which would be configured to mask potential sensitive data that are not necessary for PI operations

On the last point specifically, anonymization protocols can also be used to address regulatory compliance issues, such as GDPR compliance. Section 6 also describes how the decentralized architecture approach provides innate benefits regarding data security, regulatory compliance and separation of concern.

Overall, using the methodologies described previously will enable integration and interoperability of extensional, external and legacy data with the PI service stack. As the adoption of PI increases, it is envisioned potential marketplaces may be created for the development of such adapters. Additionally, potential system developers could benefit from creating interoperability adapters themselves to include themselves in this upcoming vertical market.

5.4.6 IoT component

The aim of the IoT component is to enable the realisation of smart containers, facilitating track and trace of containers in real-time or near real-time monitoring. To achieve that ICONET will employ IoT mechanisms and devices that will monitor PI-containers and other PI-elements. The details of IoT requirements are presented in D1.7. Initially, the plan for data requirements and IoT interoperability with the rest of the ICONET infrastructure was to use the iot-lite ontology. (Figure 2121).

Figure 21 IoT-lite Ontology Example

In this diagram, representing an example of the iot-lite ontology, a sensing device (ssn:SensingDevice) is described as part of a wider system (ssn:System). The iot-lite ontology also describes the type of the sensor (e.g. temperature), the unit (e.g. celcius), the location of the sensor (lat, long, alt), the coverage of the sensing device, as well as the service that exposes the data captured by the device. As the project and the work on the IoT cloud service has progressed significantly, it was decided that a less verbose data scheme would be used, containing the values of the IoT sensors, as well as required meta-data for the configuration of said sensors. An example of this operational message in JSON format can be seen below(Figure 22 Example of IoT operational data).

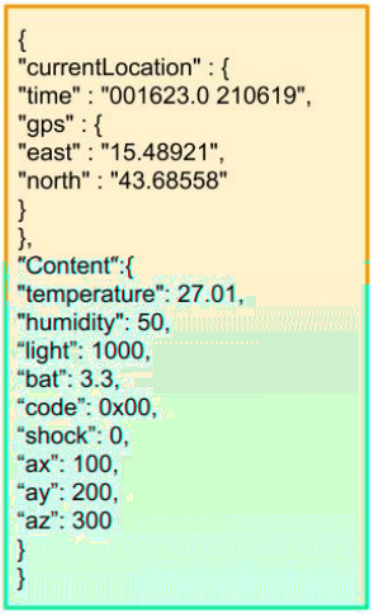

Figure 22 Example of IoT operational data

6 Physical Internet Decentralized Reference Architecture

6.1 Background of previous work

For the design of the reference architecture, ICONET has adopted an incremental and iterative approach in order to ensure that the architecture is robust and constantly updated by the evolving requirements and technology advents within ICONET and the research community in general. The outcome of the first version of this work is shortly presented in this section. The concept of Digital Twin was used, in order to position the required modules in relation to existing supply chain functions & roles. As such, the first iteration of the reference architecture, presented in Figure 23, presents a high-level view of the different components and the relations to ICONET work packages.

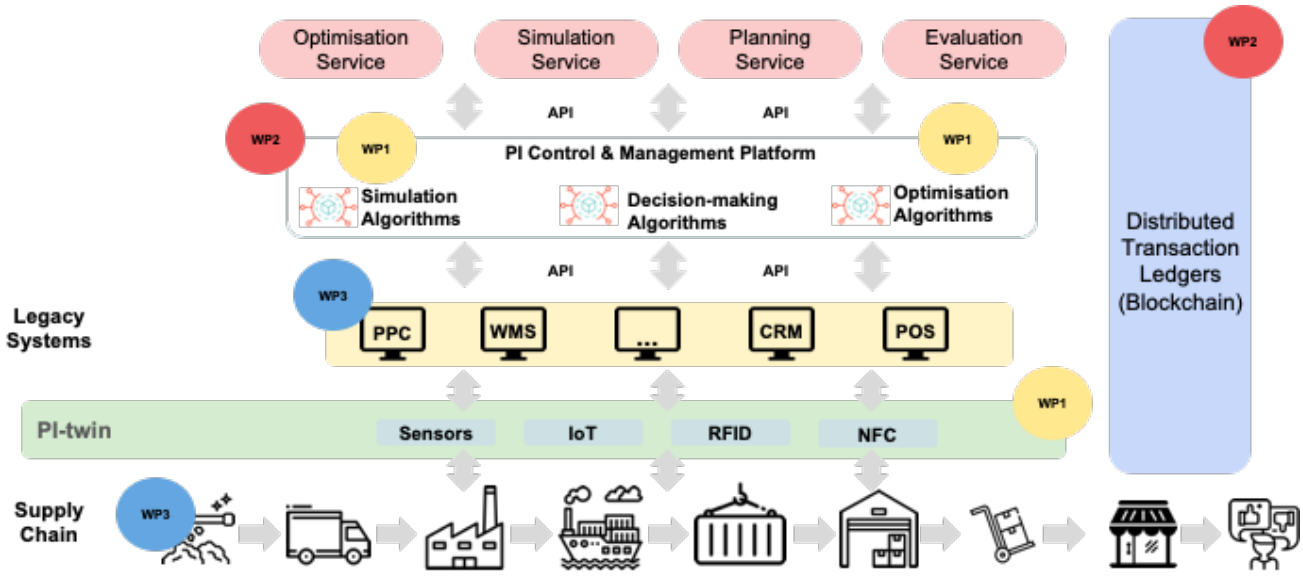

Figure 23 Initial Conceptual Architecture

For the first iteration of the conceptual architecture, an attempt was made to capture the different systems that need to be covered by the architecture and subsequently identify the data that will be available and therefore needed to be extracted, transformed and processed. In addition, all requirements stemming from the work carried out in WP1 were taken into account in order to inform the reference architecture. Table 3 provides an outline of the WP1 outputs used to inform the first iteration of the ICONET reference architecture.

Table 3 Mapping WP1 outputs

Deliverable	**Output**
D1.1 PI-aligned digital and physical interconnectivity	Previous work and existing standards on Physical & Digital Interconnectivity
D1.3 PI network optimisation strategies and hub location problem modelling	Definition of optimisation service requirements (inputs and outputs)
D1.6 Requirements And High-Level Specifications For IoT Based Smart PI-Containers	Definition of requirements for IoT connectivity

D1.7 Generic PI Case Study and associated PI Hubs Plan	Definition of general PI case study and data requirements
D1.11 PI Protocol Stack	Investigates existing PI protocols

This first iteration was further extended with research findings from relevant work, offering a more focused view of the different modules (services). The initial identified modules (Figure 24), include modules that are needed in order to support the key functionalities of ICONET.

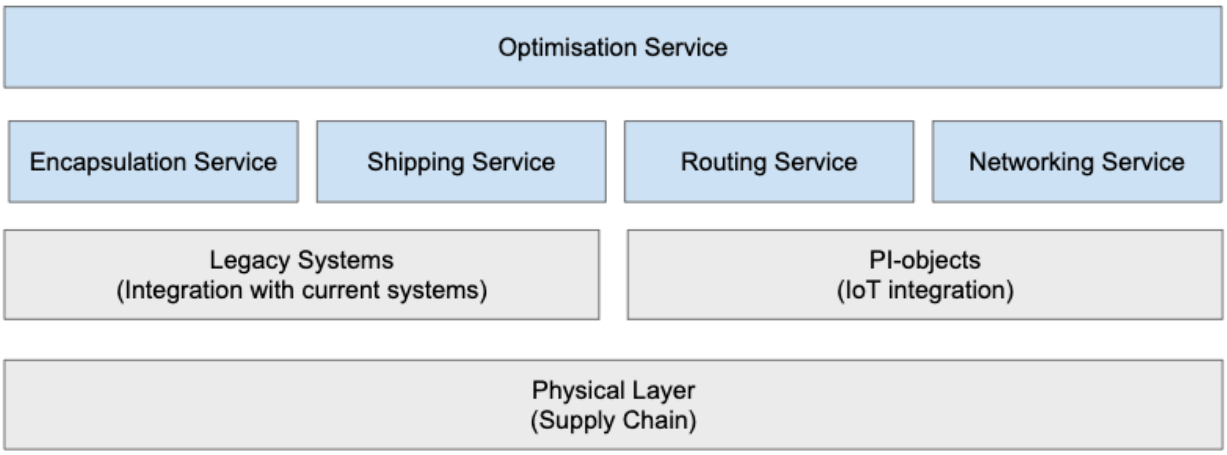

Figure 24 Key Modules

The analysis of the modules and interactions of the conceptual reference architecture were based on the findings of D1.10, and more specifically on the findings regarding the applicability of OLI in ICONET. The analysis made followed the layers defined in OLI (Montreuil et al., 2012). For the final version of the conceptual reference architecture, further developments & findings on the applicability and functions of the various layers and their corresponding services were used. More specifically, work done in D1.11 & D2.3 were used as a basis for the needs that the architecture will need to cover. The significant progress made in the technological implementation of these services and relevant discussions that were had with WP2 partners also played an important role in the final design. This further guided the conceptual ICONET reference model, where a combination of OLI & NOLI models were used to inform the final architecture. As such, the final reference architecture shifted to a more technical approach, describing the topology of services and their interactions in relation to the physical building blocks of the PI.

6.2 Current version of Reference Architecture

6.2.1 Design Principles

During the project, various workshops were organized to identify the key requirements of the services. The identification of these requirements, as presented in Section 4 of this document, helped shape the final architecture. Additional work done in WP1 and WP2 also played a critical role in the decisions made when designing the reference PI architecture. In the previous version of this document, a simple conceptual architecture was defined based on the initial interconnection requirements of the PI world with the various logistics entities, as shown in Figure 25. This diagram presents an interface between the various entities, through the use of ICONET Protocol Stack (as detailed in D1.11) and relevant ontology, with the ICONET services and PI node.

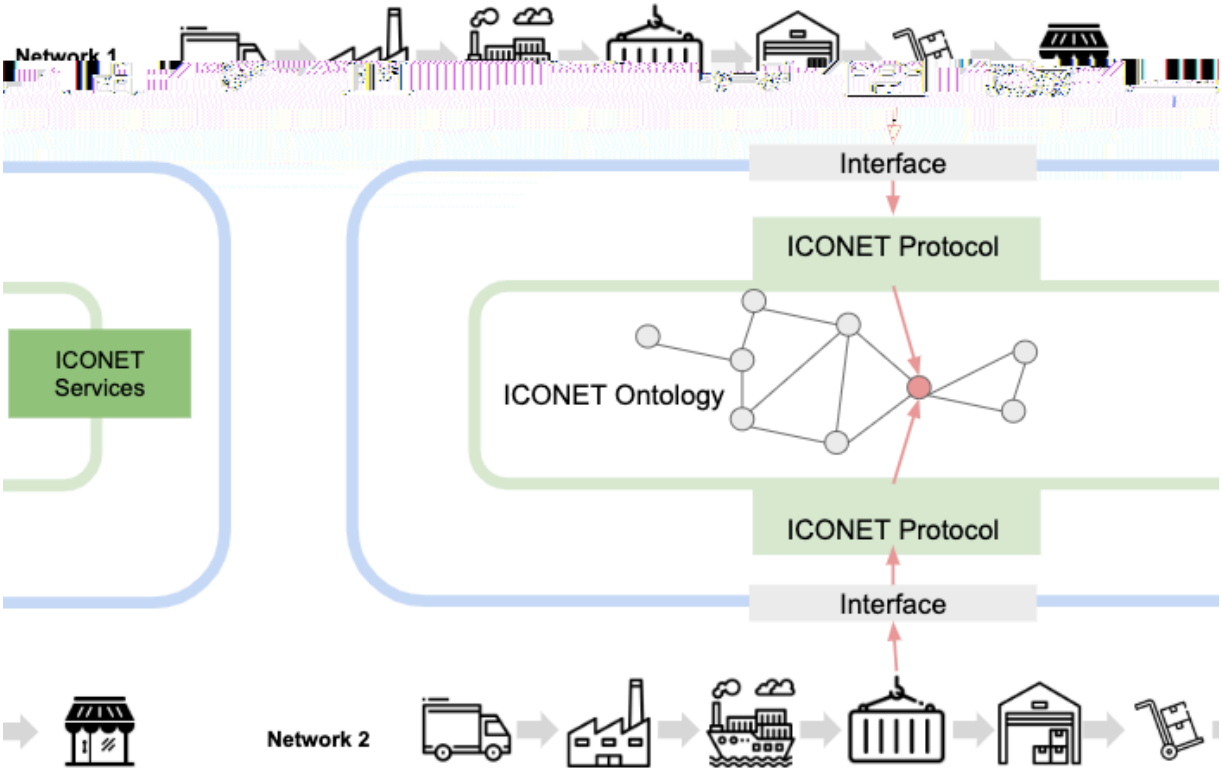

Figure 25 Conceptual Architecture

6.2.1.1 Eliciting Technical Requirements

The initial technical requirements as presented in the previous version of this report, were elicited during project wide workshops that were organized in the first period of the project. As the work progressed, the focus shifted to the design and development of the core services. The design of these services was iterated upon (as shown in deliverable report D2.4) based on requirements stemming from WP1, as well as real use case scenario requirements and capabilities needed, as was communicated to WP2 partners by the Living Lab partners. Additionally, the interconnection of these services done for the PoC Integration platform (as presented in deliverable report D2.20), spawned additional requirements that were needed to fully enable the seamless communication between them, as well as the simulation platform. During this iterative process, the PI

architecture has been continuously evolving to adapt to the newly found requirements and capabilities. The cyclical nature of the design process that was followed can be seen in Figure 26 below.

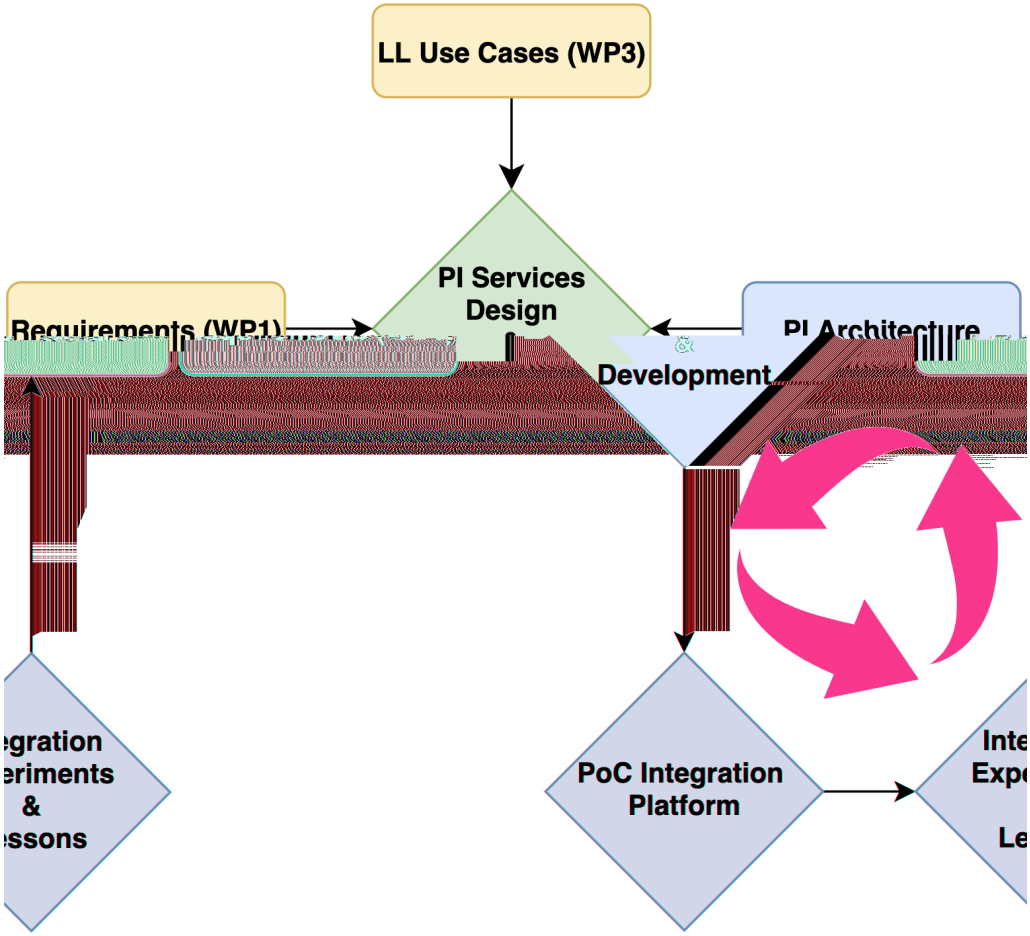

Figure 26 Cyclical influence of ICONET work

6.2.1.2 Technologies

The technologies that were used when developing the various components of the ICONET project were varied depending on the service. However, from the architectural point of view, internal technologies used by the independent services are not very relevant. Instead, the focal point of the architecture is enabling data exchange and secure communication between the components, as well as scalability and interoperation between different scopes of businesses and operations. As such, the decision to create REST API interfaces for services was made. This allows for standardized communication between services and potential different implementations of client systems, as the REST paradigm is language agnostic. Using JSON as a common data format, interoperability between the services is also achieved. Additionally, a basis for secure communications is established through using HTTPS for a standardized defence against man-in-the-middle attacks. Additional authentication capabilities are supported by the service and the proposed architecture design, such as Basic (username & password) authentication, as well as multiple Oauth2 implementations.

The data that will be generated and used by the ICONET services will grow with time and this led to discussions regarding the use of central or distributed data storage repository. The data needs of the project can be

addressed using both of these approaches. Assuming the point of the decentralization is done on the PI Hubs (meaning that each PI Hub will also act as a node for the decentralized data storage), the decentralized storage could be slow to implement, as this will require business agreements between participating parties. While the decentralized paradigm would complement the decentralized nature of the Physical Internet concept, a centralized data storage for each PI Hub would be easier to implement and set up. This means that for each party hosting a set of the ICONET services, they will also utilize a data storage repository to be used by the services.

6.2.1.3 Digital Interconnectivity

According to previous work detailed in "D1.1 PI-aligned digital and physical interconnectivity models and standards" to achieve the Digital Interconnectivity there is some previous work done to be considered as a starting point. Digital interconnectivity ensures that physical entities, constituents and factors can seamlessly exchange meaningful information across the PI, fast knowledge and fact-based decision-making in action.

The review done shows that there are many research projects and standardization initiatives but a full integrated approach for PI is not available. ICONET Project created a PI-compliant stack of models and took existing standards as basis.

The following list summarizes the analysis done in D1.1 regarding the state-of-the-art of existing projects, standards and emerging trends in the field of Digital Interconnectivity for the PI.

Data Capture & Encapsulation: the usage of GS1 Data Identification was considered and partially utilized.
Data Integration & Standard Smart Interfaces: a good work has been done in the MODULUSHCA Project, which may be extended by GS1 Data Exchange standard.
B2B Interoperability: ICONET software architecture considered distributed software systems, integrated by using loosely coupled protocols like web-services or RESTful services.
Service Orchestration & Service Choreography: to achieve an integration of distributed systems, both approaches were considered.

6.2.2 Living Lab Considerations

Apart from the design principles and generic technical requirements mentioned in the previous sections, another validation of the approach chosen for this final architectural blueprint was its applicability to the Living Labs of the project. To achieve the generic and widely applicable architecture blueprint, we can use the 4 Living Labs for validation, as they differ significantly on their scale and operations. The following Section will provide an overview of each LL with the corresponding needed information flow, as has been decided by previous work done in WP3 deliverables.

6.2.2.1 Living Lab 1 – Port of Antwerp

LL1 concentrates its efforts on the improvements of the overall railway performance within the port of Antwerp by creating a railway community for all relevant stakeholders under the coordination of the Port Antwerp Authority.

Figure 27 outlines the information flows among the key actors/platforms in Living Lab 1, along with the required PI Services. The external IoT Cloud Platform, has been included to visualize the interaction with an externally managed tracking service, feeding the PI Shipping Services.

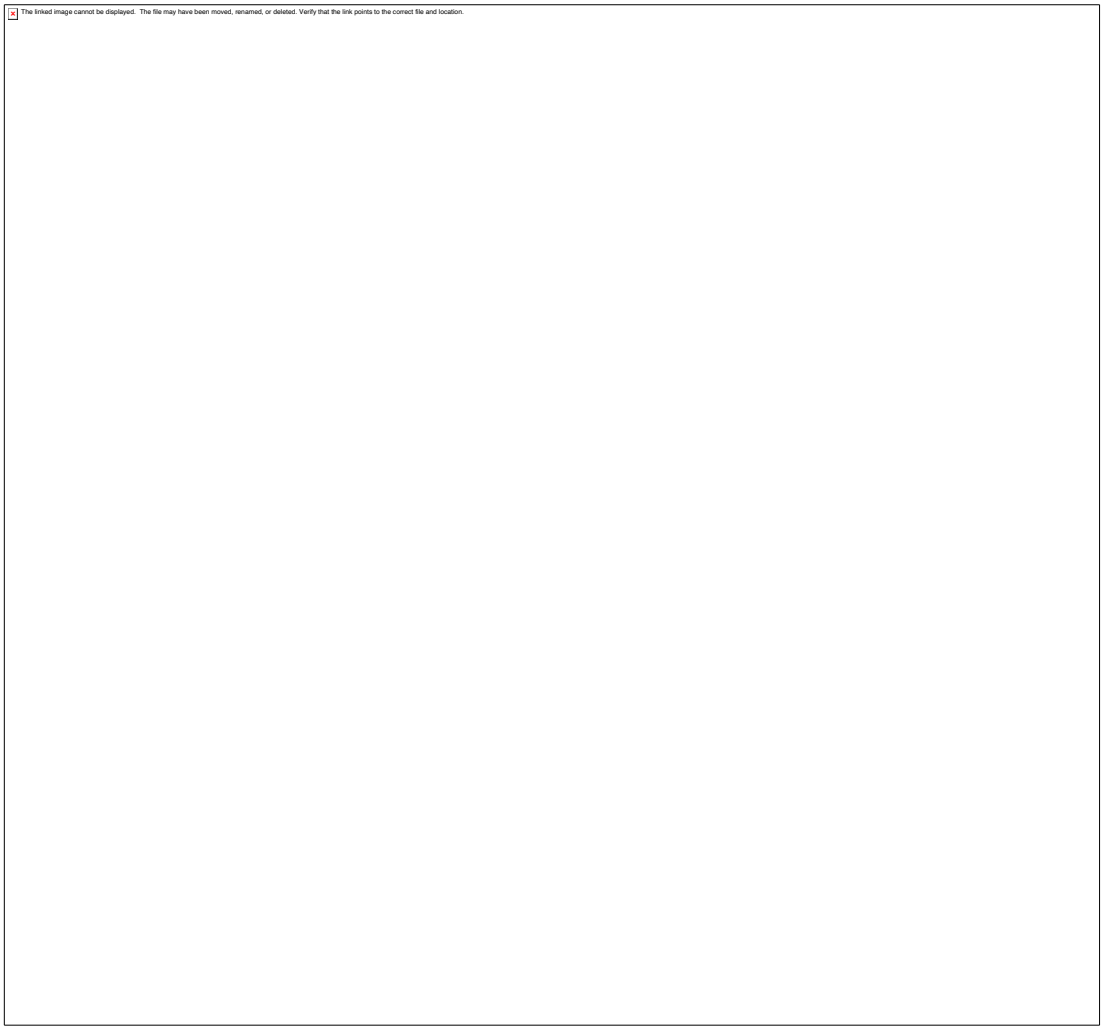

Figure 27 LL1 Service Oriented Information Workflow

This information flow helps to derive the interactions of the various services and actors in the PoA Use Case. While this is helpful in orchestrating the services, for the architectural blueprint to be applicable in this case we need to take under consideration the complexity and the physical setup of the PoA.

The PI application for the PoA will need to orchestrate shipments making optimal use of all the individual nodes & operators contained therein, as shown in Figure 28 below.

Actors	Description
Deep sea terminals (2)	MPET MSC European Terminal DPW Antwerp Gateway

Railway Undertakings (with licences)	11 in total
Combined Transport Operators	10 in total
Bundling areas (2) (used for intermediate stops)	Antwerp South Antwerp North

Figure 28 List of PoA actors

As such, the combination of the various "building blocks" of LL1 and the engagement of the capabilities and functionalities offered by the services can be seen in Figure 29 below.

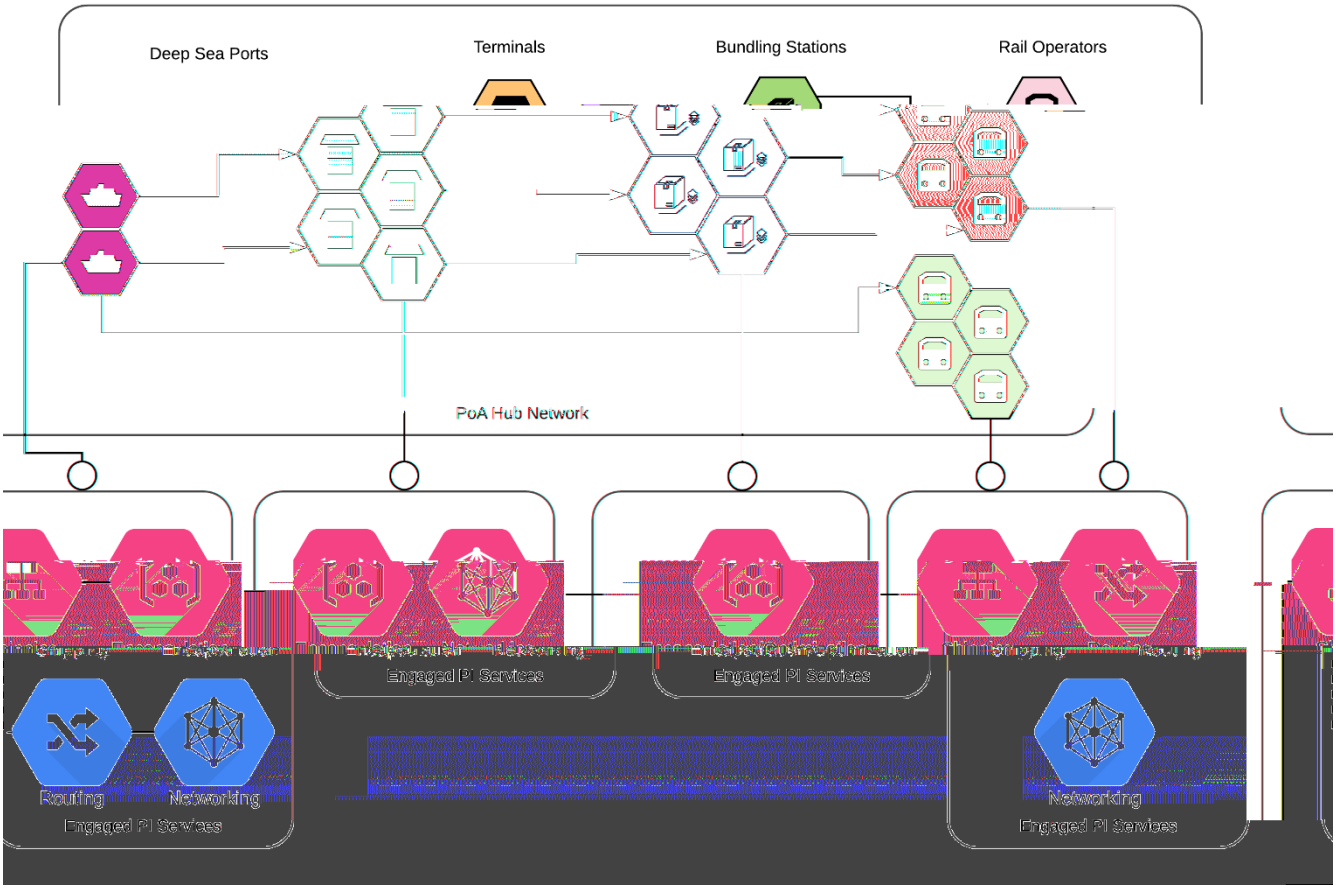

Figure 29 PI Services on PoA

In this Figure, two scenarios are examined:

1. The shipment arriving from a Deep-Sea port with an inland destination will fill the capacity of a train, and as such is loaded directly from the deep-sea terminal to depart for its destination
2. The shipment will have to be unloaded into the deep-sea terminal, sorted and moved to an inland terminal, where it will be moved to a bundling station for wagon loading. The wagons will then be assembled into a train to depart for its inland destination.

As is apparent, there is an overlap of concern for the engaged services. In more detail:

> The Deep-Sea Ports will need to engage with the Shipping Service for information about incoming and outgoing cargo Shipments. The Encapsulation service will need to be engaged for information regarding bundling and sorting operations of containers as they were loaded onto the incoming ship, or bundling and sorting decisions for cargo to be loaded into the outgoing ship. Additional Encapsulation concerns come up for bundling cargo onto a train that will be loaded directly from the deep-sea terminals.

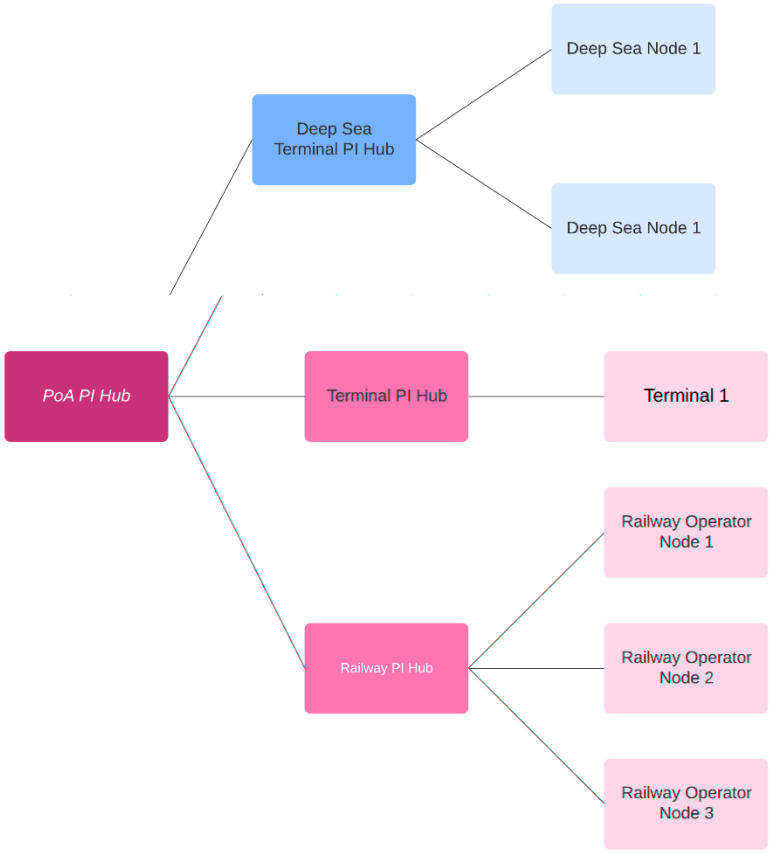

Figure 30 Decomposition of PoA PI Hub

6.2.2.2 Living Lab 2 – Proctor & Gamble

The goal of the PI application in LL2 is to provide visibility and monitoring capabilities in shipments across established multimodal corridors. The 2 corridors that are examined for this Living Lab, are Mechelen to West Thurrock using trucks and ferries, and Mechelen to Agnadello using trucks and trains.

Figure 31 outlines the information flows among the key actors/platforms in Living Lab 2, along with the required PI Services. The external IoT Cloud Platform, has been included to visualize the interaction with an externally managed tracking service, feeding the PI Shipping Services.

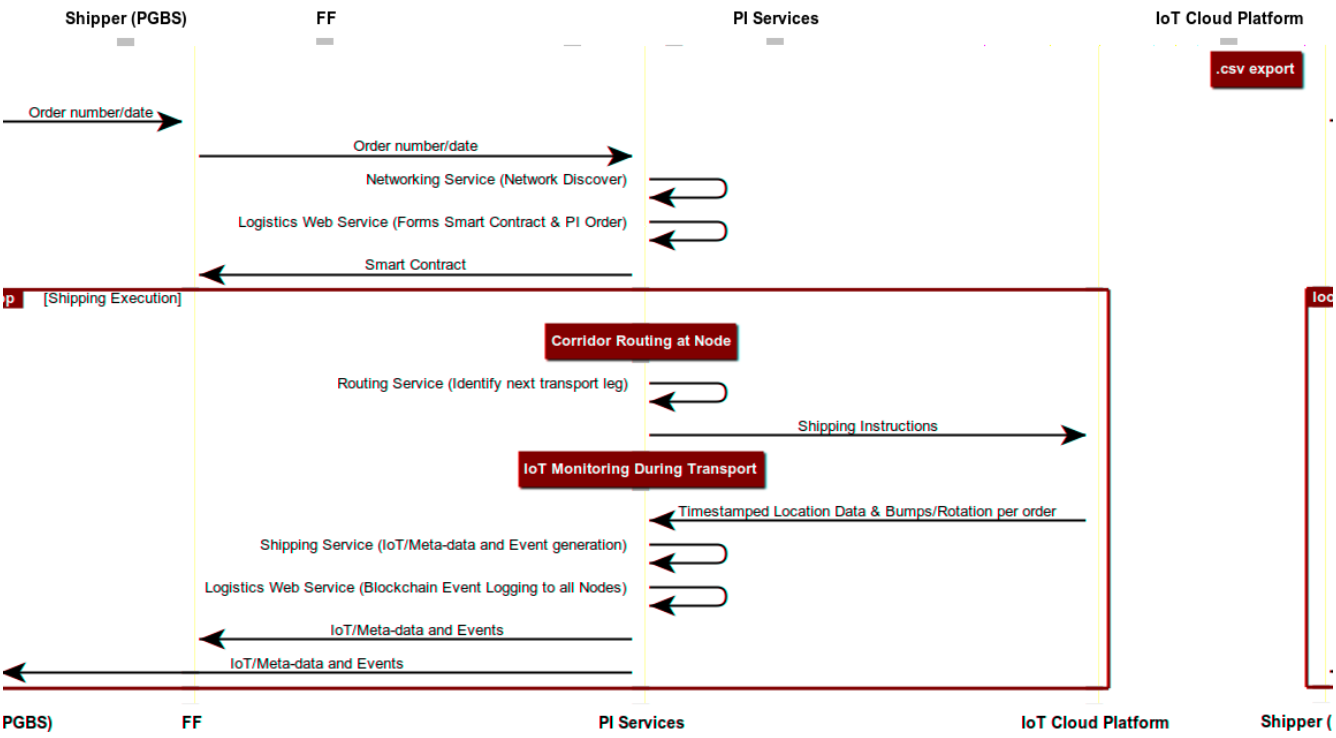

Figure 31 LL2 Service Oriented Information Workflow

While this scenario is simpler than LL1, the Track & Trace functionalities that are being utilized is one of the most important aspects of the PI concept. The PI nodes, or the "building blocks" that are utilized in this Living Lab, are essentially the origin, destination and intermediate stops of a shipment. In Figure 32 below, a high-level view of the steps taken to complete the routes using the 2 corridors are shown, in conjunction with the relevant services.

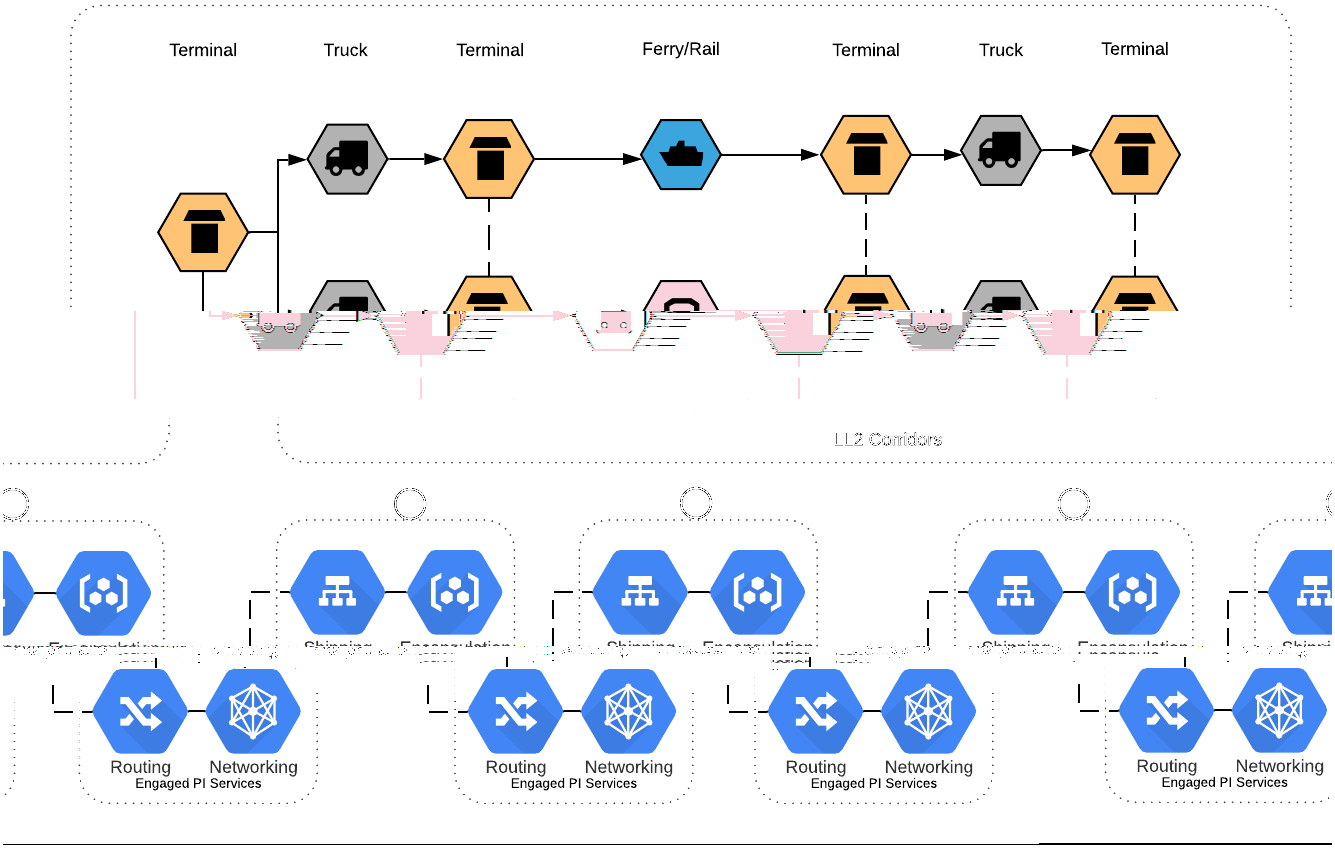

Figure 32 LL2 Corridors & PI Services

As is apparent, the PI services are engaged across every stop in these corridors. The common functionality across these nodes, in terms of service usage, can be described as follows:

The Shipping Service is present to orchestrate the rest of the services. It also provides a high-level overview of the shipment and its' state and at the same time, it receives updates from the containers containing IoT sensors through the IoT Cloud Service.

The Encapsulation service is responsible for the bundling and unbundling operations that occur in each terminal, as needed. In a potential shipment that would use a single mode of transport, this could be potentially omitted but, in this case, as is expected, bundling/unbundling operations take place before each mode change.

While the Routing service will remain mostly inactive in these corridors, seeing as the routes that the shipments take are mostly predetermined to follow the exact corridor, its' presence is important as rerouting could potentially be needed depending on the conditions of the links between two nodes.

The Networking service is also needed to have availability and capability information about these nodes, as well as information regarding the state of the links and routes between them.

The main concern for this Living Lab, as already mentioned, is to provide the tracking capabilities innate in PI across an end-to-end shipment that is using these corridors. As such, the architecture will need to position the building blocks in such a manner that runs the minimum risk of loss of information. This is achieved by positioning the Shipping Service, which is the principal tracking service, in every intermediate stop. This placement allows

for 2-way tracking of the containers and the shipment, from both the origin Shipping Service and the destination Shipping Service in any step of the Shipment.

It is also important to note, that while these corridors are comprised of individual links and routes between the nodes they encompass, they are also considered as a single route. A visual representation of this can be seen in Figure 31 below.

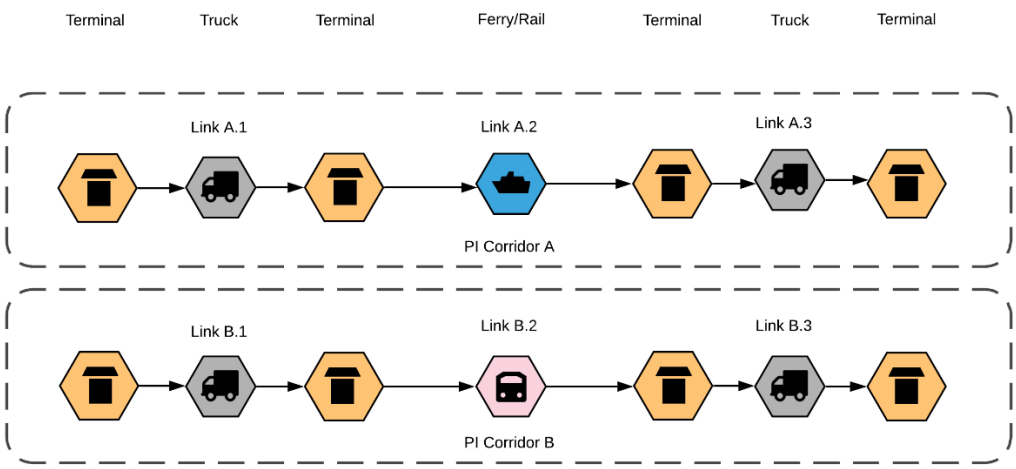

Figure 33 Links comprising a corridor

As such, when a shipment needs to arrive to the end destination of the PI corridor A for example, the initial routing will only need to route to the "PI Corridor A" initial hub. The following routing and shipment through the corridor will be then performed by the internal corridor services residing on individual hubs. This means that the initial routing, in conjunction with the networking service, can consider a PI Corridor as a single link, with corresponding information, such as mean travel time, condition, expected delays etc. referring to the unified corridor instead of each single individual link.

6.2.2.3 Living Lab 3 – SONAE

LL3 will address the challenges of eCommerce channel fulfilment in urban environments and study the role of the regional distribution centres-hubs as PI nodes within the PI paradigm. In particular, SONAE will investigate on how regional warehouses jointly with local stores (smaller) can act as **PI Nodes** for urban distribution and fulfilment of eCommerce Purchase Orders, with the main goal of reducing stock-outs, costs and reducing lead times.

Figure 34 outlines the envisioned information flows among the key actors/platforms in Living Lab 3, along with the required PI Services. The image below shows an outline of the initial approach to the connection between the different services and the simulation model. The model begins with a set of orders, one sender wants to send through PI network. Using the networking service, a coordinator can configure the nodes and transports which are available in the network to manage the selected orders. The simulation model can also be interfaced with the route optimization service to determine the best urban transport routes to deliver the orders to the customers with the selected strategy.

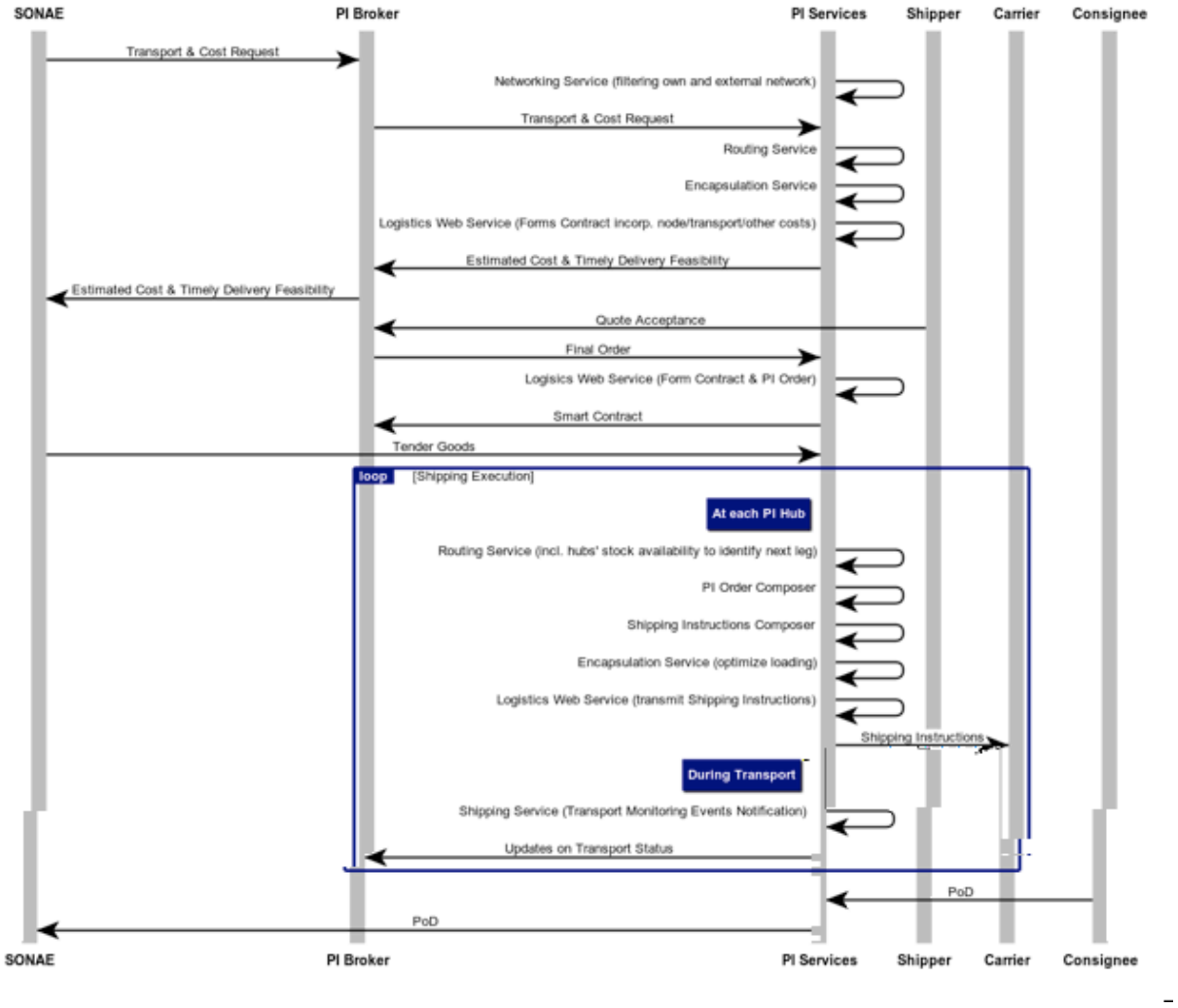

Figure 34 SONAE Urban Distribution Workflow

Apart from the interconnection of the services, Figure 35 below presents the high level flow between the main components of this Living Lab.

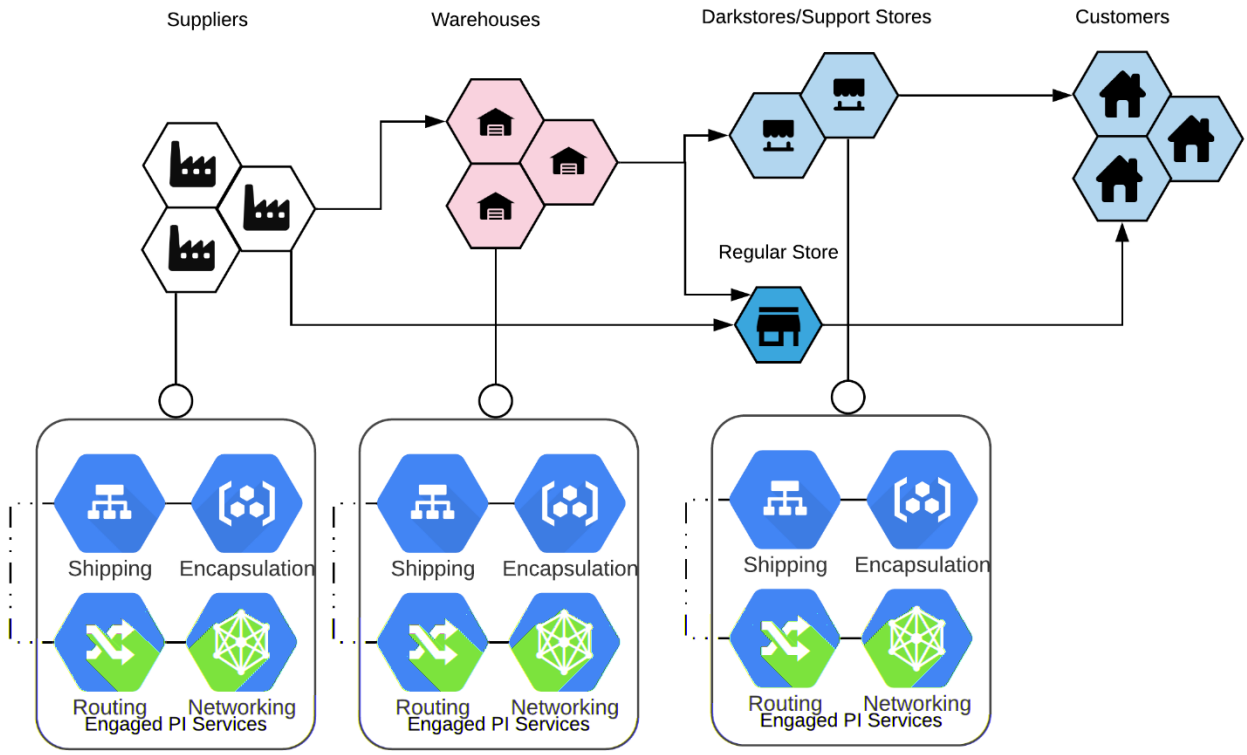

Figure 35 SONAE Flows & Engaged PI Services

While some similarities exist with the previous Use Cases, this Living Lab is quite different. As can be seen in Figure 35, the engaged PI services exist across most of the PI Hubs & Nodes that comprise the operations of LL3, with the exception of regular stores and last mile delivery destinations. The key difference stems from the goal of this Living Lab, which is reducing stock outs and lead times. This means that the transfer of goods between the involved nodes needs to be optimized. For this reason, ICONET examines the possibility of additional internal orders being generated to replenish stock across stores. Again, the common functionality can be summarized as follows:

Shipping Service will need to engage with Suppliers, Warehouses and stores where possible to provide better visibility of incoming and outgoing shipments. This can be particularly useful across the board in this LL, as it will allow for better tracking of incoming and outgoing stock of products in stores.

Encapsulation service is needed in every location where loading & unloading operations occur, meaning across Suppliers' warehouses, central SONAE warehouses, darkstores, support stores and retail stores. The Encapsulation service will be tasked with optimally loading the products that are to be distributed in an order that makes sense and streamlines delivery and unloading operations.

Routing service will need to route the various shipments across the different nodes, from origin to destination, and as such is needed on each node where shipments can originate from or conclude to. Additionally, a Rerouting action can occur when a delivery truck needs to load products to be delivered from another facility than originally planned, due to stock-outs.

Networking Service is needed across the various nodes, as it needs to be up to date regarding available stock, by using the information tracked by other services and external systems to monitor the incoming and outgoing stock of products, based on orders.

Taking into account what was previously mentioned, it is clear that the problem that the PI concept is called upon to give a solution to, is the internal orchestration of stock traffic, to reduce stock-outs, as well as position the stock in stores closer to the end customers, to reduce delivery times. These two optimizations would obviously reduce costs associated with transport and customer support that would be needed to address these stock outs and offer replacements.

To accommodate this, the PI reference architecture will need to be flexible enough to allow additional internally orchestrated orders to be generated, originating from within the PI services instead of the classic paradigm that positions the PI orders as originating externally. These dynamic, data-driven generated orders will greatly optimize an eCommerce Network, as replenishment actions can be triggered and orchestrated automatically. Of course, the level of automation will need to be configurable by the interested parties and be set within desired limits.

6.2.2.4 Living Lab 4 – Stockbooking

LL4 is designed to investigate the potential of e-Warehousing as a key enabler of the PI concept. LL4 will provide warehousing services structured around the PI concept, which will be tested and enhanced in the LL.

This section describes the technical integration performed and the next simulation dealing with PI network. It will start by defining the test scenarios. Then describe the algorithms planned to simulate the PI and lastly defining the data management of the system.

Figure 36 outlines the information flows (To-Be) among the key actors/platforms in Living Lab 4, along with the required PI Services. The external IoT Cloud Platform, has been included to visualize the interaction with an externally managed tracking service, feeding the PI Shipping Services.


PI Web Logistics Service
PI Encapsulation Service
PI Routing Service
PI Network Service
WaaS
PI Shipping Service
Cloud IoT Service
Order
PI Order & Assigned PI Container
Request Routing Table
Network Discovery
Request Warehouse Availability/Conditions
Warehouse Availability/Condition
Network (incl. warehouse availability)
Routing Calculation
Optimal Warehouses
PI Order
Calculate Optimal Warehouses
Route
Shipping Instructions
Form & Expose Shipping Instructions
Shipping Instructions
Establish Smart Storage Contract
During Transport & Storage
Request IoT Data
IoT storage/transport events & conditions

Figure 36 LL4 Service Oriented Information Workflow

Apart from the interconnection of the services, as shown in the Figure above, this Living Lab, similar to Living Lab 3, focuses on the dynamic allocation of stock across Stockbooking warehouses. Unlike LL3, this allocation originates from customer orders and is not entirely decided by Stockbooking itself. However, the goals of the PI offered optimizations for this Use Case are alike. Reducing stock-outs, allocation of stock in different warehouses (per customer request) and optimization of the warehouse fulfillment rate. As such, the PI services will offer the same dynamic data-driven decisions to automate and optimize decisions made on products originating from the Suppliers, or moving cargo between the Stockbooking warehouses.

6.2.3 Final PI Architecture Blueprint

Based on the extended study of PI Reference models, the functionalities of services and their interactions, as well as their inputs/outputs and data requirements, the design approach mentioned in the previous section and the Living Lab considerations, the current version of the PI reference architecture was formulated. The main challenge was to find a solution that can accommodate the efficient operations for various logistics actors who participate not only in different scales, but different operational modes. The PI architecture will need to cover logistics operations that occur through a city, but at the same time cover operations that occur through countries. Additionally, multiple modes of transport need to be covered by a generic solution. Moreover, the dynamic nature of some of the cases examined will need to be accommodated by the PI architecture and the overall service functionality.

Similar to LL1, Figure 37 provides a simplified example of the different scales of operations. In the example, a shipment produced by a factory, needs to reach a deep sea port for global transport. It would need to be initially transported locally to a PI hub that will support transport across a region (national or international). Assuming the regional transport destination is not the deep sea port, it will need an additional local transport to reach it. In this very simple example, the shipment will go through 3 hubs that operate on a local, regional and global level respectively. Additionally, 3 modes of transport are used: trucks, trains and freighters.

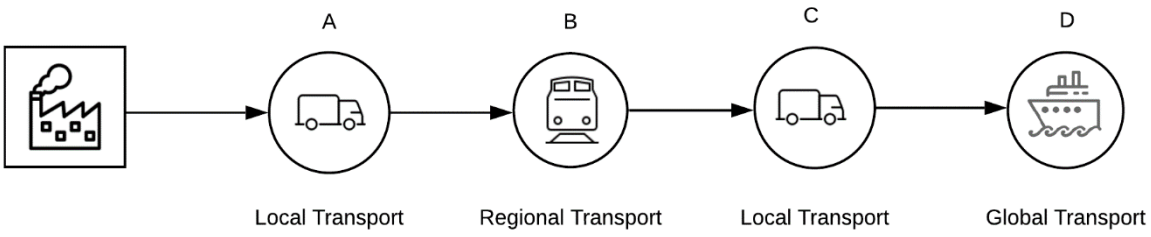

Figure 37 Example of different levels of transport accommodated by PI hubs

The PI architecture needs to support each of those hubs, their level of operations and the different modes of transport. In a centralized architecture, this would mean that the centralized repository of the ICONET services would need to have data available that span a vast geographical area, while also maintaining and updating routes, locations, and link conditions for all the potential nodes that reside therein. In contrast, in a decentralized paradigm, each node will only have to contain information about the operations pertaining to the specific node.

Moreover, based on the needs of LL1, it is apparent that the physical installation of the services doesn't always correspond with the physical operations that need to take place for the fully enabled PI flow of goods. As such, it is important to note that the stack of PI services can either be installed using local hardware or even be hosted in the cloud, as long as their operations concern a single node or hub. Additionally, some of the building blocks will need to be engaged by one or two of the services, and not the whole stack. This means that the services will be built using a plug-&-play approach, while the instances of the services they depend on will be configurable, allowing for single service installations that communicate directly with services located in other nodes, if needed.

Based on the conclusions made from LL3 and LL4, regarding the dynamic nature decisions in some networks, the architecture as well as the services themselves have already been positioned in a manner that allow a single service to initiate an order. Assuming that these decisions are based on stock-outs, the Networking services would be the most appropriate one to initiate these orders, since it has knowledge for stock levels across each node.

As such, following the concept of the PI, where software and networking concepts and technologies are applied to the logistics world, the decision was made to follow a decentralized paradigm for the PI reference architecture, with PI nodes acting as nodes of the system. Decentralized systems in the modern world offer many advantages in comparison to centralized systems. System availability on a decentralized system is not dependant on the health of a single node, as if one node fails others can continue operating independently, essentially eliminating the single point of failure as would be the case with a centralized system. Scaling of resources can be done vertically for individual nodes, allowing them to add more resources according to their operations' needs. Furthermore, the autonomous nature of a decentralized system is a great fit for ICONET and the PI concept, as services residing on each node allow these nodes to configure their instances as they see fit and will not have to rely on a central authority or service provider for specific implementations.

The decentralized paradigm offers great capabilities in terms of service interactions and relevant data. The services residing in a single node need to only contain and access information about operations pertaining to that single node, eliminating complex data stores. This information is then made available when requested by another node in the network.

As is apparent, this also has ramifications in the data security department. Node operators will be more likely to allow data stemming from external and proprietary systems to be made available to the PI network when having the capability to host these services themselves locally. As mentioned in section 5.4.5, this data can be transitioned to the PI network by using the PI Data adapters for external systems which would also be hosted in the same installation as the PI services, allowing greater control of when, how and which data should be made available to the PI network.

To better visualize the implementation of the decentralized architecture (based on the points raised above) , the PI services and their interactions are presented using the PI nodes as a basis in Figure 38 below:

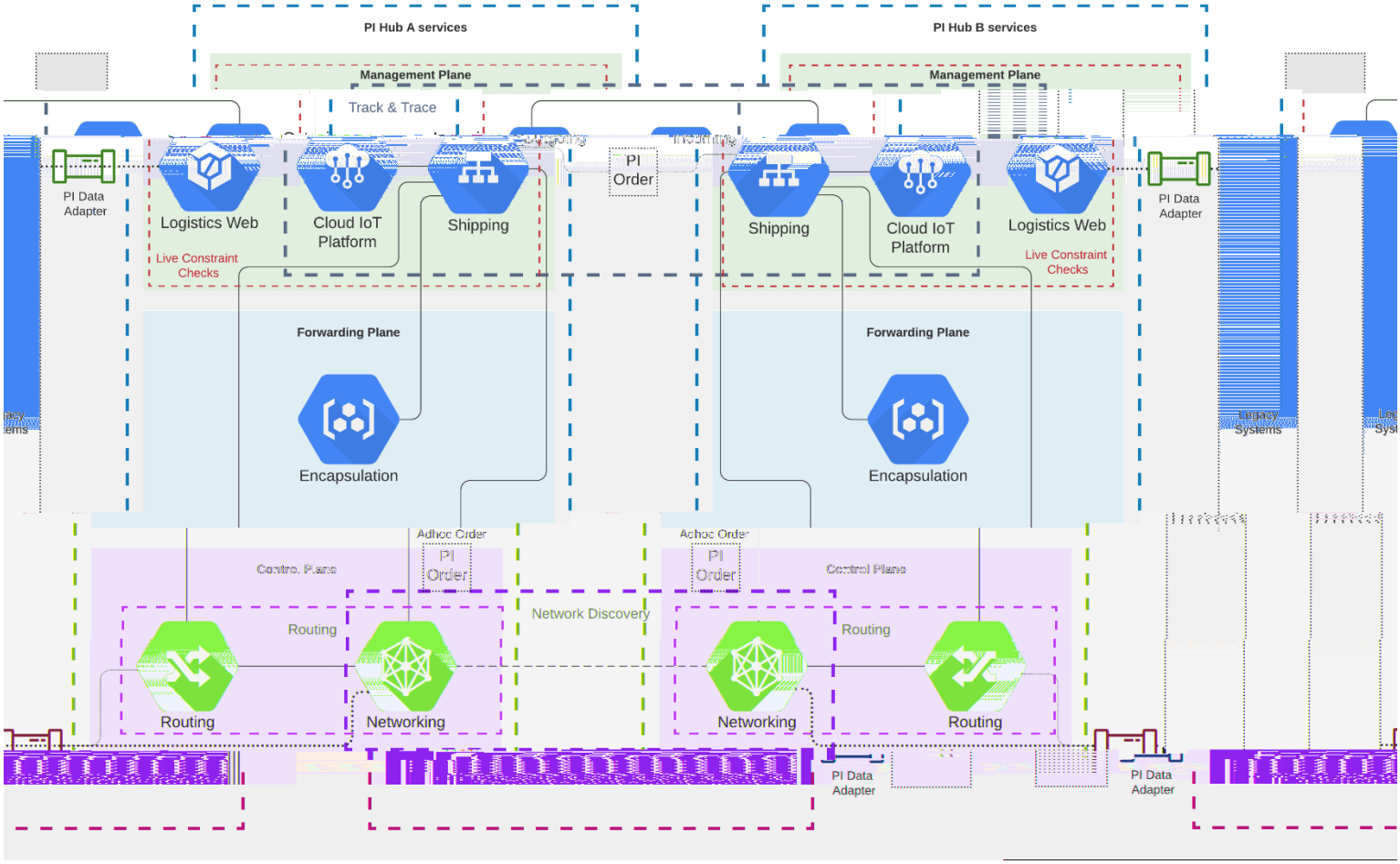

Figure 38 Node to Node communication

This architecture diagram shows the interactions between services on a single node, as well as communication on a node-to-node basis, with shared interactions between services being highlighted. Figure 39 below shows the same internal interactions positioned along with physical logistics elements, showcasing the applicability of the architecture in scenarios with the same goals and needs as the Living Labs described in the previous sections.

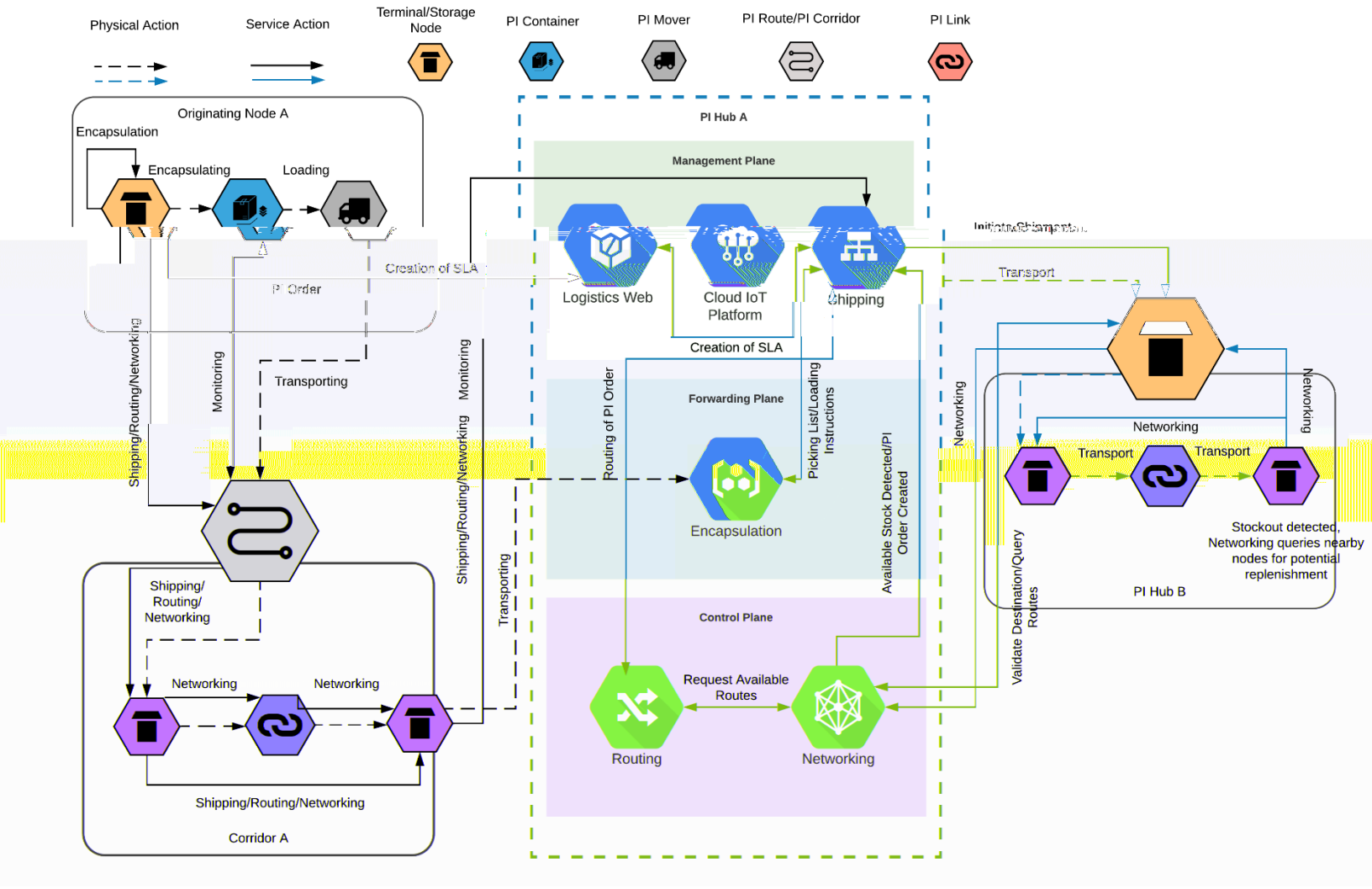

Figure 39 PI Reference Architecture

This figure examines two scenarios (note that all nodes contain their own set of PI services):

In black, flow of an order passing through a corridor and its' internal links and stops.

In blue, flow of an order originating from a composite PI hub due to an internal node Stock out, with no option of local replenishment

For the first scenario, a shipment originating from a PI node with its' own set of services is routed to Corridor A. For this to happen, the shipment is first encapsulated into PI containers and loaded onto a PI mover. The originating node services communicate with the initial node of the corridor to notify it of the incoming shipment. This is done with the service flows described in previous sections. Similarly, to LL2, this initial corridor node then utilizes its' own stack of PI services to communicate and route the shipment to the corridors' internal nodes. After the shipment reaches the final corridor node, the node communicates with the services residing in PI Hub A to the incoming shipment, after which the routed and potentially further encapsulated cargo is transported to PI Hub A.

For the second scenario, a sub-node of PI Hub B (similar to the multilevel nature of LL1) detects a stock-out using the Networking service. This sub-node then queries its' neighbouring nodes for potential replenishment options.

As no such options are available, it finally communicates with PI Hub A, which can fulfil the stock replenishment. A PI order is then generated from PI Hub A, which follows the PI service flows (creating an SLA, encapsulating products and routing them to PI Hub B). The order then is transported to the central PI Hub B node, after which the central node, using its' own services, routes the shipment to the sub-node. This scenario is similar to the dynamic nature of LL3 & LL4, as described in the previous section.

The overall functionalities, in both cases, can be split in three distinct planes, Control, Management, and Forwarding. A detailed overview of these planes can be found in the following section.

6.2.3.1 Management Plane

In computer networking, the management plane of a networking device is the element of a system that configures, monitors, and provides management, monitoring and configuration services to, all layers of the network stack and other parts of the system.

The management plane of the PI Node and its services is responsible for orchestrating, managing, configuring and monitoring a shipment through its lifecycle. As can be seen in Figure 21, this plane contains the Logistics Web, Shipping, and IoT Cloud services. The overall management of a shipment will happen through these services, either directly through the PI via the Shipping Service, or potentially even through a legacy system through the connecting Web Logistics service. All of these services contribute to maintaining the quality of a shipment intact, by continuously querying each other to monitor sensor values. Additionally, the Track & Trace functionality of the PI is enabled by two of those services. This allows the shipping services of both nodes to receive live updates on the state of the shipment, while checking (as mentioned previously) against constraints on SLAs stored in the Web Logistics service. This allows both origin and destination to better orchestrate their overall operations, greatly reducing intake and outtake times. Furthermore, in case of an impeachment, both origin and destination are notified and can take appropriate actions.

6.2.3.2 Forwarding Plane

In computer networking, the forwarding plane, sometimes called the data plane or user plane, defines the part of the router architecture that decides what to do with packets arriving on an inbound interface. Most commonly, it refers to a table in which the router looks up the destination address of the incoming packet and retrieves the information necessary to determine the path from the receiving element, through the internal forwarding fabric of the router, and to the proper outgoing interface(s).

In the PI, the forwarding plane of a node contains the encapsulation service. This service is responsible for encapsulating the incoming shipments to PI containers and potentially encapsulating these containers further into other containers or directly into PI movers, essentially enabling the forwarding of the shipment to other nodes.

6.2.3.3 Control Plane

In network routing, the control plane is the part of the router architecture that is concerned with drawing the network topology, or the information in a (possibly augmented) routing table that defines what to do with incoming packets. Control plane functions, such as participating in routing protocols, run in the architectural control element. In most cases, the routing table contains a list of destination addresses and the outgoing interface(s) associated with them. In the PI node, the Control plane contains the Routing and Networking services. The networking service is responsible for holding all information of the node that needs to be made available to other nodes, describing capabilities, potential routes, capacity, locations etc. This allows the Routing

service to make better decisions as to where to route a potential shipment to, as the Networking service of the candidate nodes can provide important information. As such, the Networking service of one node can exchange information with the corresponding service of another node, and similar to network routing, create a "routing table" to be used by the routing service from the data exchanged between the various nodes. This can be done through Network Discovery, essentially the communication between Networking services of nodes, or can be directly fed information from legacy/external systems.

6.2.3.4 Relation to NFV and SDN Network Paradigms

NFV, or Network Function Virtualization is a relatively new conceptual network architecture, that uses IT virtualization techniques to virtualize entire classes of network node functions into individual building blocks (Virtual Network Function or VNF Components) that may be used to connect, chain together, or create communication services. Essentially, NFV describes a way to reduce cost and accelerate service deployment for network operators by decoupling functions like a firewall or encryption from dedicated hardware and moving them to virtual servers. Instead of installing expensive proprietary hardware, service providers can purchase inexpensive switches, storage and servers to run virtual machines that perform network functions.

While NFV is not directly associated with the PI and ICONET project, some of its' innate benefits were used as an inspiration for what the architecture should achieve. In a more abstract sense, the concept of PI, which is equating classical logistics functions and decisions to corresponding IT components and services, is similar to that of the NFV, as they both take traditionally hardware or physical related roles and virtualize them to be performed via software methods. Of course, the main difference being, that the PI still has a physical element to it, as no matter the decisions made by the software components, corresponding physical actions still must be taken. However, offerings of the NFV concept, such as modularity, scaling and virtualization of functions is covered in the ICONET architecture. The services are positioned in such a way that they can be substituted or removed entirely if a PI node is not interested in a specific function of the PI. For example, if a PI node operator does not perform additional bundling operations in their premises, the encapsulation service would not be used. The distributed nature of the architecture as mentioned previously, offers great capabilities for scaling up or out. Additionally, the virtualization of functions occurs throughout the PI services, as again in a more abstract way, they perform virtual actions to reach conclusions that will later be translated to physical actions.

Software-

It is important to note that while both paradigms can be used in conjunction, they are not interchangeable or dependent on one another. Their principles complement each other, but there can be implementations where only one of both is used. The feasibility and relation of these network paradigms has been also explored in D1.11 and D2.19, where an application of SDN is implemented in relation to the PoC platform.

6.2.3.5 Conclusions

In conclusion, the presented PI reference architecture is sufficiently generic and high level to encompass a variety of requirements, as expressed previously. Both multilevel and dynamic use cases were considered and addressed, based on real world use cases from the ICONET Living Labs. NFV and SDN network paradigms were examined in relation to PI, with some of their design patterns being utilized for the reference architecture. Interfacing with external systems can be done through the use of the PI Data Adapters, as described in section 5.4.5. Communications between external systems and the PI stack, as well as the internal communications of the PI stack will be sufficiently secured by using current industry standards for data protection, security and privacy. Regulatory compliance is also achieved by utilizing some of the same standards. In addition, the decentralized paradigm followed has a clear separation of concern that occurs naturally, as services are positioned locally instead of centrally, which also emphasizes data sovereignty across various operators and parties. The result is a widely applicable decentralized PI reference architecture, that provides value to the PI concept as it describes a simple yet robust architectural paradigm that can be adapted to most logistics operations and can be iterated and built upon as the PI concept becomes more widespread.

7 ICONET Ontology

Ontologies are knowledge representation tools that have advanced through the years and are currently used in a extensive range of applications across different fields. The most widely used definition is the one coined by (Studer, Benjamins & Fensel 1998), who defined ontology as "a formal explicit specification of a shared conceptualisation". This definition is a product of the evolution of the initial definition coined by (Gruber 1991), who defined ontologies as "vocabularies of representational terms – classes, relations, functions, object constants – with agreed-upon definitions, in the form of human readable text and machine enforceable, declarative constraints on their well-formed use.".

Ontologies are essentially a group of terms organised in a hierarchical structure (class-subclass), describing a specific domain (Trokanas, Cecelja & Raafat 2014). Ontologies are enriched with properties characterizing terms and restrictions on these properties. Finally, ontologies are completed with instances representing specific entities of the domain.

The data that need to be exchanged will be represented by an ontology. The aim of the ICONET ontology is to enable interoperability and standardisation among different systems and services, hence eliminating any syntactic or semantic heterogeneity. The ICONET is based on the common data model for PI (as developed in the MODULUCSHA project) (Figure 40), however the ontology development process followed a bottom-up approach, starting by modelling data identified in Figure 41 (Sternberg and Norman 2017). The initial ontology is presented in Figure 40.

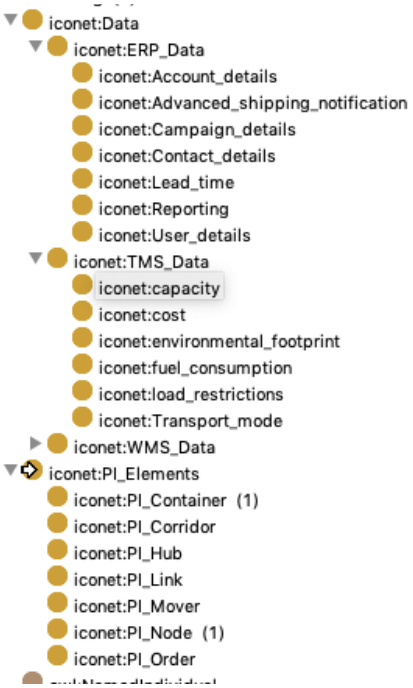

Figure 40 ICONET Ontology excerpt

Building on the common data model (Figure 41) and extending it, the ICONET ontology also covers sensor data.

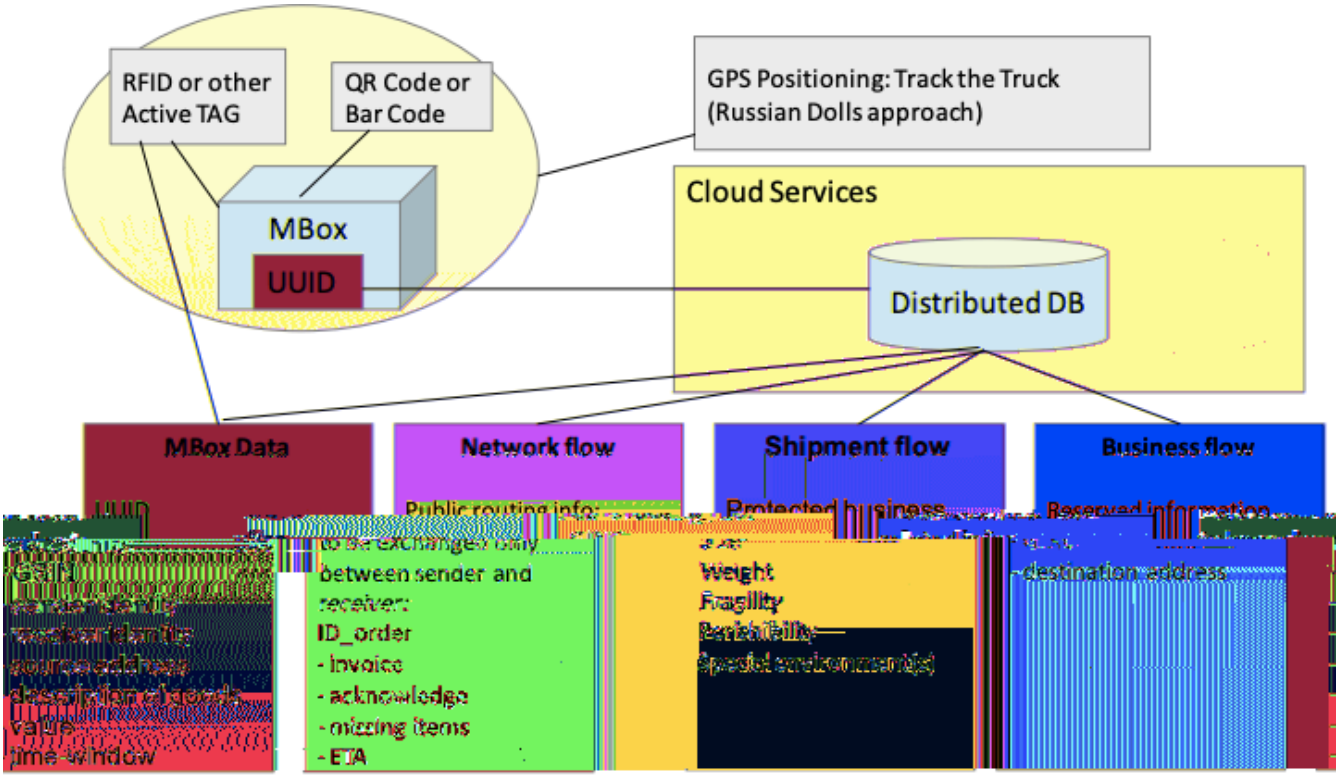

Figure 41 PI Common Data Model [Sternberg and Norrman 2017]

The ICONET ontology builds on existing knowledge and is expanded to include semantic rules that can capture relevant knowledge and use it to evaluate data as well as certain aspects of operations. For example, semantic rules can be used to flag cases where the temperature of a product in-transit has exceeded the permitted limit and needs to be pulled out of the network. Such an example rule is presented in Figure 42.

```
# Batch Compromised Due to High Temperature
ASK WHERE {
   ?this disco:hasMaxTemperature ?maxtemp .
   OPTIONAL {
      ?this disco:hasDispensingHistory ?dh .
      ?dh disco:hasTemperature ?dhtemp .
      FILTER (?maxtemp < ?dhtemp) .
   } .
}
```

Figure 42 Example semantic rule

To achieve that, the data model covers all aspects of an end-to-end supply chain and their digital representation. Additionally, ontologies have been recognized as a tool for achieving interoperability between systems and IoT devices. The complete ICONET Domain Model is presented in Figure 43 below.

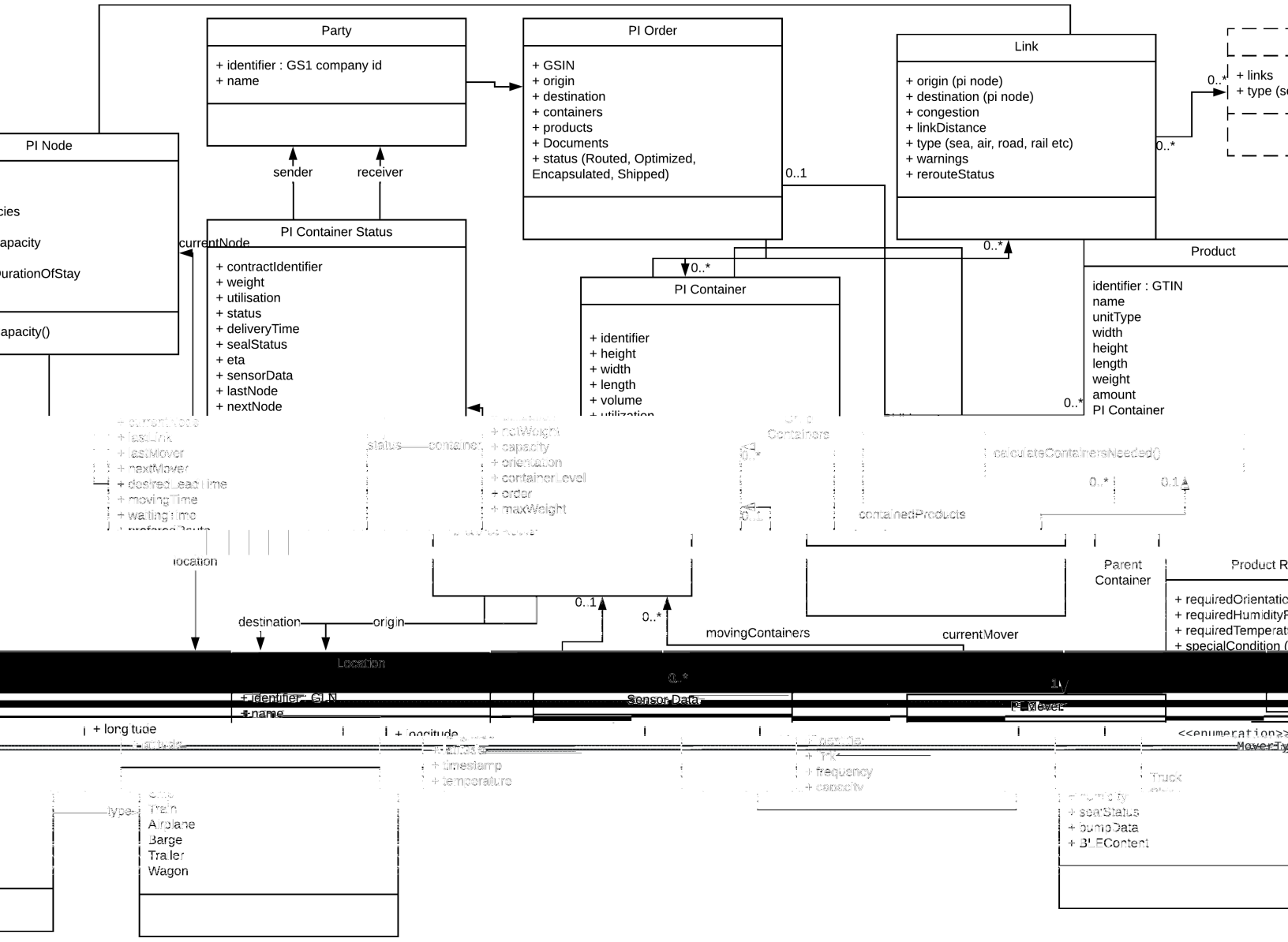

Figure 43 ICONET Domain Model

Domain Model, as shown, above, models the PI entities that are of interest to the key services. For the IoT sensors, a different model, l ns mentioned earlier, was chosen. As data size in sensor communications is important, this minified model does not contain PI semantic ICONET Domain Model through the PI order and the Shipping Service. Figure 44 shows an initial PI order and the relevant IoT section to be co oud Platform to configure the various sensors and corresponding alerts.

```
"PIOrder":{
// other meta data
 "CloudIoTService":
 {"Shipment":{"carrierCode":"iconet","tripCode":54,"orderID":"12AbCxYz","containerID":"00291939"},
 "Requirements":{
  "temperature":{
    "basicConfiguration":{"activation":"true","unit":"C","sendMeasurement":"true","sendAlarm":"true",
    "advancedConfiguration":{"numberOfAlarms":3},
    "alarmConfiguration":{
      "alarm1":{"max":50,"min":0,"sendAlarmIfOutsideRange":"true"},
      "alarm2":{"max":100,"min":70,"sendAlarmIfInsideRange":"true"},
      "alarm3":{"min":-5,"sendAlarmIfOutsideRange":"true"}}},
```

The corresponding acknowledgement from the IoT Cloud Service is shown in Figure 45.

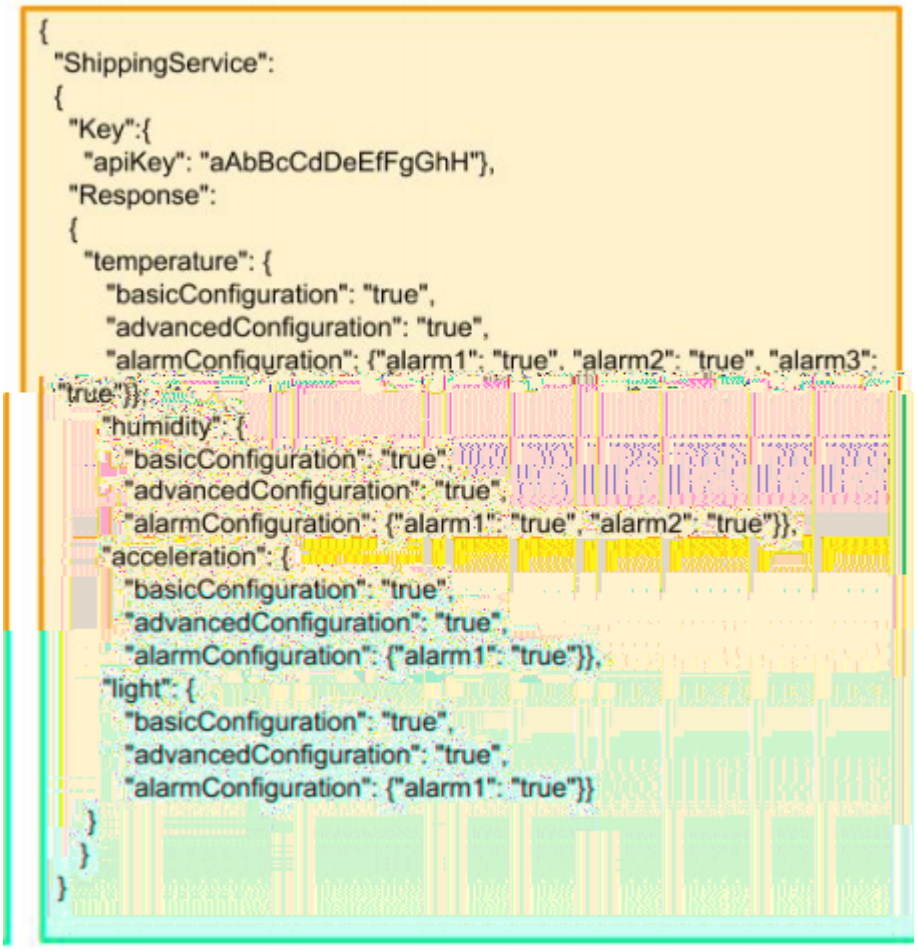

Figure 45 Acknowledgement of PI Order & IoT settings

The specifics of the IoT operations and the corresponding Data Model are described in detail in D2.8.

8 Conclusions

This deliverable provides a final blueprint for a PI enabled decentralized architecture, which will serve as a steppingstone towards the realization the Physical Internet. It introduces the ICONET Reference Model, which is comprised by the OLI & NOLI layers. This reference model is then associated with the service specifications. A consolidation of the requirements stemming from the key services, their interoperability and the data specifications needed for their operations is then presented, accompanied by an example which demonstrates their roles, as well overarching flows and interconnections between them in a PI enabled shipment scenario. The integration with external and legacy systems is also addressed, introducing the PI Data Adapters, which through various implementations can be used to achieve interoperability, while also maintaining data security and protection and regulatory compliance.

The work done previously on the PI reference architecture, the overall design approach and the Living Lab considerations is presented as a segue to the final PI Reference Architecture. Based on the design principles and requirements analysed before, as well as NFV and SDN technologies, a final high-level, generic and widely applicable blueprint is presented as a decentralized system, with each PI Node serving as a node hosting its' own set of services. NFV and SDN technologies, and their corresponding design and principles, are presented and further analysed with relation to reference architecture. Furthermore, the allocation and separation of services into the Forwarding, Control and Management planes is presented and explained. A common data model (an ontology) has also been presented for organizing the required data and enabling interoperability, leading to the final resulting ICONET Domain Model and IoT data models.

The experiences and lessons learned that were applied in making this architectural blueprint, will shape future implementations of the Physical Internet. While the widespread acceptance and adoption of the technologies and operations described are still quite far, the work done in ICONET and the resulting architecture, which was shown to be able to function in a variety of scales and operations, will be useful in future considerations of the technological implications of making the Physical Internet a reality.

9 References

Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., & Taylor, K. (2017). IoT-Lite: a lightweight semantic model for the internet of things and its use with dynamic semantics. Personal and Ubiquitous Computing, 21(3), 475-487.

Gruber, T.R., 1991. The role of common ontology in achieving sharable, reusable knowledge bases. KR, 91, pp.601-602.

Mertzanis, K. (2016). Designing Protocols for the Physical Internet (Dissertation).

Montreuil, B, E. Ballot, F. Fontane. *An Open Logistics Interconnection model for the Physical Internet. P*roceedings of the 14th IFAC Symposium on Information Control Problems in Manufacturing Bucharest, Romania, May 23-25. 2012. Available: https://www.sciencedirect.com/science/article/pii/S1474667016331718

Sternberg, H., & Norrman, A. (2017). The Physical Internet–review, analysis and future research agenda. International Journal of Physical Distribution & Logistics Management, 47(8), 736-762.

Studer, R., Benjamins, V.R. and Fensel, D., 1998. Knowledge engineering: principles and methods. Data & knowledge engineering, 25(1-2), pp.161-197.

Trokanas, M. Bussemaker, and F. Cecelja, 2016. Utilising Semantics for Improved Decision Making in Bio-refinery Value Chains. Computer-Aided Chemical Engineering, 38, pp.2097-2102.