iQonic

Innovative strategies, sensing and process chains for increased Quality, re-configurability, and recyclability of Manufacturing Optoelectronics

Deliverable

D3.5 Smart tagging for automatic setting-up the process chain parameters

Deliverable Lead: SENSAP

Deliverable due date: 30/09/2020 (M24)

Actual submission date: 10/05/2021

Version: 1.0

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 820677

Document Control Page	
Title	D3.5 Smart tagging for automatic setting-up the process chain parameters
Creator	SENSAP
Description	This deliverable will document the prototype Implementation of the Smart Tagging application
Contributors	SENSAP, BRUNEL, IP-ASCR, CORE, ALPES
Creation date	15/03/2021
Type	REPORT
Language	English
Audience	☒ public ☐ confidential
Review status	☐ Draft ☒ WP leader accepted ☒ Coordinator accepted
Action requested	☐ to be revised by Partners ☐ for approval by the WP leader ☐ for approval by the Project Coordinator ☒ for acknowledgement by Partners

REVISION HISTORY

Date	Version	Author	Comments
15/03/2021	0.1	SENSAP	Initial structure of document
21/03/2021	0.2	SENSAP	Contribution to chapter 1
19/03/2021	0.3	SENSAP	Contribution to chapter 2
02/04/2021	0.4	SENSAP	Contribution to chapter 3
16/04/2021	0.5	SENSAP	Contribution to chapter 4
23/04/2021	0.6	SENSAP	Contribution to chapter 5
27/04/2021	0.7	SENSAP	Ready for peer review
07/05/2021	0.8	ALPES	Peer review by Tobias Gresch / ALPES
08/05/2021	1.0	FRAUNHOFER	Final Review

Table of Contents

Table of Contents

DISCLAIMER

EXECUTIVE SUMMARY / ABSTRACT

This report (D3.5 Smart tagging for automatic setting-up the process chain parameters) deals with the definition, design and implementation of the Smart Tagging system. The Smart Tagging System tends to a) transform a product to a smart product according to industry 4.0 specifications following its life cycle and b) to provide a mechanism for automatic machinery setup during the production phase.

Special pocket-less gel-assisted die trays, that are used to carry and transport sensitive optoelectronic devices between different production stages of an optoelectronic product, are selected as smart devices. Each die tray is divided to several cells, where optoelectronic devices can be mounted and transported. Each cell can be used for a limited number of mounts. The smart product will contain all information about its health status and furthermore, the Smart Tagging System will be able to suggest the best mapping available every time a die tray is going to be filled with optoelectronic devices, extending this way its lifetime.

In parallel, the die tray is used to feed the assembly line with optoelectronic devices. Each time a die tray is put in the machinery, the Smart Tagging System provides the necessary information to the machinery in order to recall device mapping inside the die tray from the ALPES logistics System. This approach reduces the setup time for the machinery.

Note that the present version of the deliverable represents the design and the prototype implementation of the Smart Tagging System that was tested in lab environment. During WP7 “Integration & Testing – Validation” the system will be further developed, receiving feedback from the integration with other components and – furthermore – from the use cases implementation.

SCOPE

The scope of this deliverable is to describe the adaptation of Smart Tagging technologies to the optoelectronic industries, and more specifically, the usage of it to the intermediate production stages of optoelectronic products. It targets to the devices that are commonly used to temporary store and transport intermediate optoelectronic devices between production line to assembly line.

ABBREVIATIONS

API	Application Programming Interface
BAP	Battery-Assisted Passive
EEPROM	Electrically Erasable Programmable Read-Only Memory
HF	High Frequency
HTTP	HyperText Transfer Protocol
IC	Integrated Circuit
IP Code (IP-XX)	Ingress Protection Code
LF	Low Frequency
NFC	Near-Field Communication
PVC	PolyVinyl Chloride
RF	Radio Frequency
RFID	Radio-frequency identification
RHCP	Right Hand Circular Polarized
RP-TNC	Reverse polarity – Threaded Neill–Concelman
RTLS	real-time location systems
SDK	Software Development Kit
UHF	Ultra-High Frequency
UI	User Interface

1 Introduction

Nowadays, the demand for optoelectronic devices is rising and the advances in optoelectronics technologies require modern, data-enhanced and industry 4.0 compliant approaches in order to meet the requirements of small-lots and high-variants productions of the sector, thus contributing to achieve right-first time, efficient and sustainable opto-electronics process-chains.

The Smart tagging systems focuses on the challenging field of the opto-electronics process chain. The production of optoelectronic devices is highly distributed among different factories and productions stages. An optoelectronic product is composed of various devices. Usually, the production line of these devices (such as semiconductor lasers) is placed in a different location from the final assembly line. Special pockets are used to store and transport these devices to the next manufacturing or assembly line. These pockets that are called die trays can mount – depending on their size – a specific number of devices. Their storage area is divided in numbered cells and each mounted device can be easily discovered based on the cell-s address-es that it is mounted on.

1.1 Objectives

The main objective of the Smart tagging system is to convert die trays to smart objects monitoring their life cycle in order to:

- Maximize their usage over time.
- Provide to the machineries the necessary information to reduce its setup time.

1.2 Relationship with other Tasks

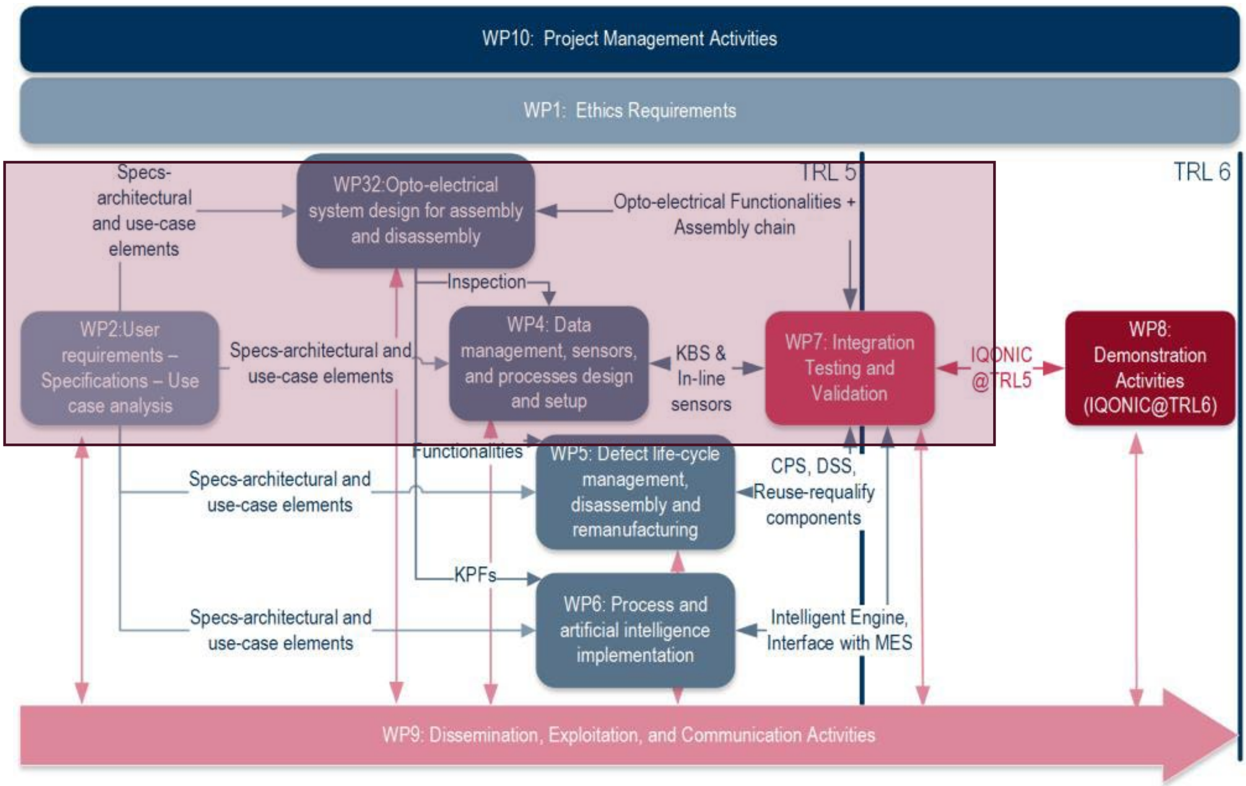


Figure 1: WP relationship diagram

The present document is closely affiliated with the following work packages and tasks (Figure 1):

WP2 - Requirements and Conceptual Process Chain Design

- T2.1 – Consolidation of user requirements
- T2.4 – Use case design

WP3 - Opto-Electrical System Design for Assembly and Disassembly

- T3.4 Inspection & metrology infrastructure to monitor the KPIs

WP4 - Data management, sensor, and processes design and setup

- T4.1 Multisensorial data acquisition network for manufacturing opto-electrical parts

In order to gain a better understanding of the production process of each use-case, so as to better identify the specifications for the Smart Tagging system, we followed the on-site visits to the use cases Alpes Lasers, Prima Electro and to technology provider ficonTEC. Through the observation of each process, as well as the discussion with the experts among our partners, the problem statement is addressed and described.

1.3 Document Structure

This document is structured in the following way:

- Chapter 1 contains an **Introduction** of the deliverable scope and structure. It provides an overview of the role of this task along with its relationships with related work packages.
- Chapter 2 presents the **Problem Statement** concerning this document and provides the basic concept behind this along with the scenario definition. The requirements and specifications are presented in this chapter as well.
- Chapter 3 provides the **RFID Theory and Hardware Specifications.** Theoretical background information concerning the RFID technology is presented here along with the hardware specifications and prerequisites that will guide the hardware selection.
- Chapter 4 describes the specific **System Architecture and Implementation** of the incoming inspection of semiconductor devices on a wafer level within the ALPES use case in the iQonic system. It provides a description of the device and data communication with the iQonic sensorial network along with system use-cases descriptions.
- Finally, Chapter 5 presents the **Conclusions** and the next actions related to task T3.5.

2 Problem statement

2.1 Basic concept

This paragraph explains the main concept of the selected approach. The first approach was tending to convert each individual optoelectronic device – that is used to assemble the final product – into a smart product, containing all the relevant information during its life cycle (production, testing, transportation to the assembly line, and finally embed to the final product). The goal of the smart tagging system was to provide to each individual device the functionality of a "smart product". After analyzing the process of optoelectronics production, the smart tagging was not able to be applied directly to the parts of the product. The main limitations were i) the size of these parts that many times are small enough to fit any type of tag over it, and ii) the fact that RF readers are not common at all in photonics systems for a readout of sub-assembly and component's properties. Also, optoelectronic industries have already produced traceability systems that monitor intermediate sub-products during production process.

The final approach is focused to the transportation of these devices from one production stage to another. Many times, these stages are not placed near each other. As an example, the production of the semiconductor laser may be placed on a totally different factory than the final laser assembling. Due to the size and sensitivity of these devices, a special equipment is required to carry these devices from a location to another one. These are pocket-less gel-assisted die trays. We will refer to them as die trays.

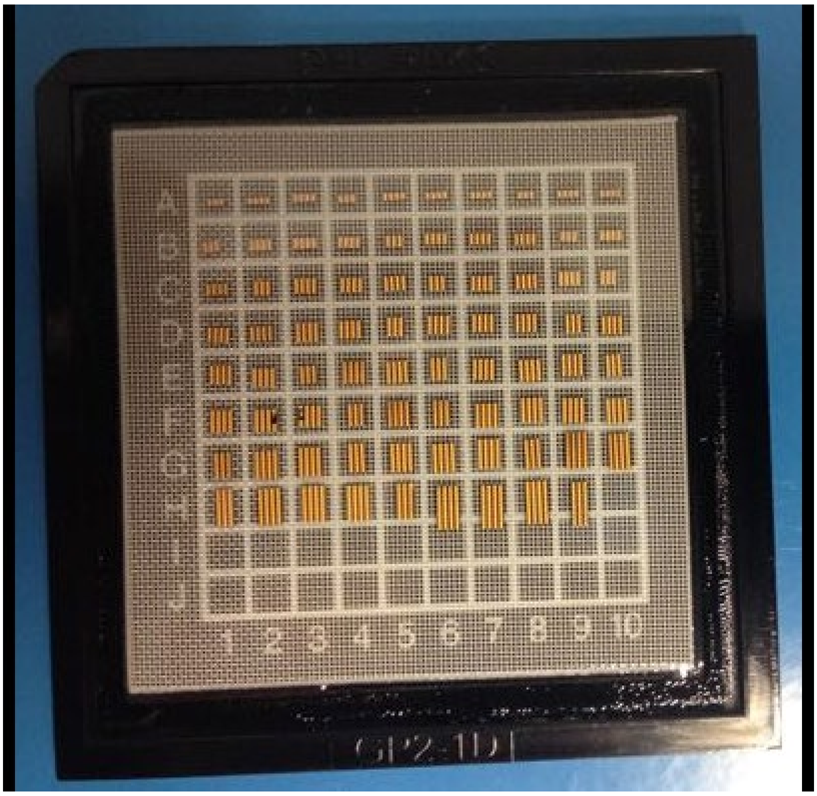


Figure 2: An example of die tray storage area

Die trays are special caskets that are made using a proprietary gel material that effectively immobilizes a device using only backside surface contact. A die tray can carry one or more devices safely, from one production stage to another. A typical die tray is depicted in Figure 2 and consists of a grid or list of cells where optoelectronic devices can be stored. Of course, die trays cannot be used infinitively: Each time a device is put on a cell (which is referred to as die tray location), it will leave some particles that will diminish the adhesion of this cell. After a number of usages, this will result in cells not being able to carry other devices anymore. One of the smart tagging system scopes, is to achieve a uniform usage of each individual cell. As a result, the die tray utilization and lifetime can be maximized.

Our basic concept is to convert these die trays to smart products and monitor the life cycle from the initialization (first usage) till the end of its life. The smart tagging system will be focused on the health status of each die tray and will update this information during transportation processes between production phases. Furthermore, with the use of the smart tagging system, the necessary information can be provided to the machinery in order to extract the location of each device inside the die tray. This can be very beneficiary as it speeds up the machinery setup time.

2.1.1 Die tray types

Several types of die tray are used to carry optoelectronic devices. The main characteristics of the die trays are:

- the cells layout. The layout can be in the form of a grid or a list. In our use case, grid-based die tray will be used (while being flexible for other types of try layouts also)
- the number of cells. There is a variety of die trays: from 5 cells to more than 100 cells. In our use case we will only use 10X10 die trays (while again being flexible for other types of try layouts)
- the numbering of the cells. The numbering depends on the layout. If the layout is a grid, there are two types of numbering: a) the row is represented alphabetically and b) the columns numerically
- the size of each cell in (x,y)
- the distance between the borders of the die tray and the cell layout. (x,y)

The full variables for each die tray type are defined into the ALPES system. Since we are going to use grid-based die trays with a specific number of cells (10X10), the most important characteristics are the dimension of each cell and the numbering of each cell. In our use case, we will use the following types of die trays:

- Die tray with alphanumeric grid notation and 100 cells (10 columns and 10 rows) (Figure 3, left picture)
- Die tray with transposed alphanumeric grid notation and 100 cells (10 columns and 10 rows) (Figure 3, right picture)

Figure 3: 10X10 with alphanumeric grid notation. Left: regular and right: transposed notation

The difference between these two die trays is that for the first type, the grid position of each cell is defined as following: a) for row definition we use a letter (A, B…J) and for column definition we use a number. The other type uses Letters for column definition and numbers for row definition. In both die trays, the cell location is defined with the same way: <Letter><number>. For example, F4. If we want to define a range of cells, we define the top left cell of the area and the bottom right one, as shown to the Figure 4:

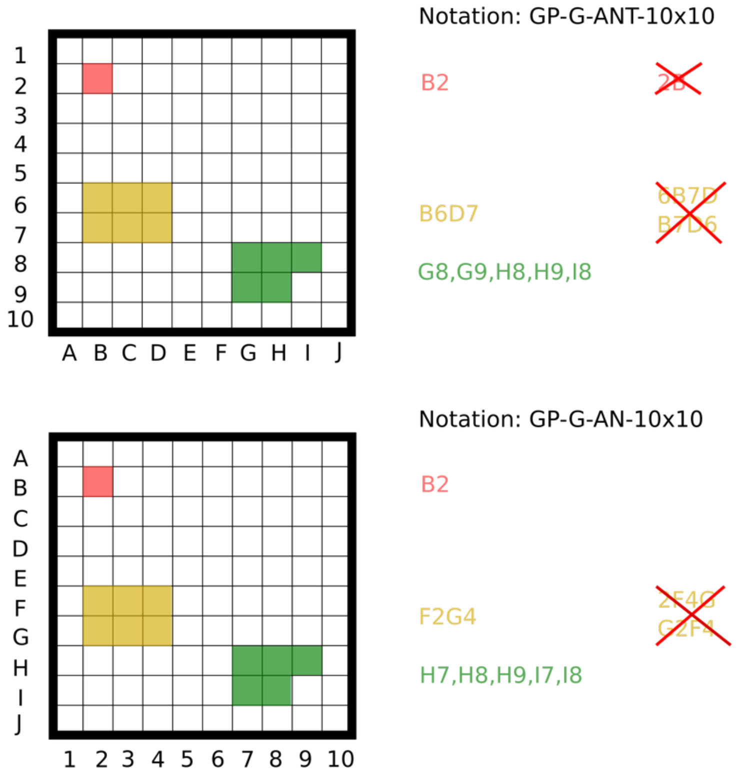


Figure 4: Location definition in the die tray (source ALPES System specification)

2.2 Scenario definition

In order to better understand the functionality of the smart tagging system we defined a set of die tray usage scenarios.

- **Initialize a new die tray**. When a new die tray is going to be used, smart tagging system needs to initialize it. The initialization process is focused on a) the definition of the maximum lifetime duration for which this die tray can be used, b) the definition of the maximum usage of each cell inside the die tray and c) providing the pilot's tracing system with a unique id that identifies the specific die tray. After the initialization, the die tray is ready to be used. Also, the initialization process is done once for each die tray. All necessary information that is related to the die tray life cycle, is stored to a RFID tag, and is attached to the die tray.
- **Mount a die tray with devices**. Once a die tray is initialized, it is ready to receive a mission (be filled with optoelectronic devices and transport these devices to the destination that is another production stage). Every healthy and empty (or idle) die tray can be used in this scenario. The tracing system requests the smart tagging application to "suggest" a location mapping for a set of devices that are going to be put inside the die tray. The smart tagging application reads the health index matrix from the RFID tag and based on a mapping strategy, suggests the best positioning for these devices. The scope of the smart tagging system is to uniformly reduce the health index of each cell in order to avoid "bad" areas. This can increase the duration where the die tray can be useful. The machinery at the production stage fills the die tray with the devices and – when it finishes – updates the smart tagging applications with the locations that were used. The smart tagging application decreases the health index for each cell that was used. The die tray is ready to be moved or shipped to the next production stage.
- **Unmount devices from die tray**. Since a die tray is filled with devices, it can be used in this scenario. When a device needs to be retrieved from a die tray, the smart tagging application reads the embedded to the die tray RFID label. The machinery is informed for the die tray appearance and extract the Location mapping of devices from the tracing system. Then, the machine extracts the appropriate

device (or devices) from die tray. This process can be performed multiple times until the die tray becomes empty. After this, the die tray can return back.

- **Read a die tray and/or manual definition of health index**. In order to include the case of the die tray fell-off during transport (invalidating the location for storage) or with visible traces, a user through a UI can flag the cells that are damaged. For the next usage of it, the algorithm that calculates the new location mapping will not use these damaged cells.

The scenarios above cover the usage of die tray into the transportation of optoelectronic devices between different production stages. By monitoring and managing the health status and lifetime duration of each die tray we can achieve their maximum usability (cost reduction) and minimize the machinery setup time. These scenarios will be described with more details in Chapter 4 "System Architecture and implementation".

2.3 Requirements

The smart tagging application was defined based on ALPES use case requirements. Our approach was as abstract as possible, in order to apply the smart tagging system to other use cases as well. In this paragraph we depict requirements that derived from the specific use case.

ST_REQ_01	**Monitor and update health index for each cell and the remaining lifetime of die tray:** In order to convert the die tray to a smart product, the proposed solution must store and depict the health status of die tray and each individual tray.
ST_REQ_02	**A unique identifier for each die tray.** Each die tray should be tagged with a unique identifier for the whole duration of its lifetime.
ST_REQ_03	**Interfacing with ALPES system**: Smart tagging system will not carry any information that are related with the devices that a die tray carries. An ALPES "die tray" logistics system stores the device mapping inside the die tray. The smart tagging system will provide the unique identifier for each die tray.
ST_REQ_04	**Suggest a location map for mounting the die tray with devices:** The smart tagging system should suggest the best locations for inserting new devices into the die tray.
ST_REQ_05	**Provide a user interface:** This UI should provide all basic interactions with each die tray (define a new die tray, mark an area as damaged, depict health status etc).
ST_REQ_06	**Support online and offline mode to the update of die tray:** The machinery that fills the die tray with devices, will probably not be located near to the Smart tagging system. The health index update should be done in offline mode. If Smart Tagging system can be installed to the machinery the update process can be done on the fly.
ST_REQ_07	**Communication with the machinery:** The Smart tagging system should communicate with machinery that puts new devices to the die tray and send the identifier of the die tray.
ST_REQ_08	**Support multiple scenarios.** Support the "*Fill a die tray with devices*" and "*Retrieve devices from die tray*" scenarios simultaneously.

2.4 Specifications

Since smart tagging system is a combination of software and hardware solutions (components) we distinguish the specifications in two categories: hardware and software. To this paragraph software specifications will be defined and analyzed. Hardware specification will be defined in the next chapter.

SW_SPEC_01	**RFID tags for each die tray.** RFID tag will be used for the conversion of die trays to smart product. Each tag will store information related to the die tray (health status, expiration date, type etc). The smart tagging application will read/write information from the tag. (ST_REQ_01 and ST_REQ_02)

SW_SPEC_02	**Communication with RFID system:** Based on ST_REQ_01 and ST_REQ_02, the application should interface with an RFID system. The selected system will provide an API or an SDK. The application will communicate with it through a read/write controller.
SW_SPEC_03	**Read data from the tag.** The application can read all available tags that are in the reader range. Also, for each tag, it can read the health status data for the related die tray. Expands the SW_SPEC_01.
SW_SPEC_04	**Write data to the tag.** The tag occurs during the initialization scenario (initialize a new die tray) and during mounting scenario (update health index table). The update process should be provided and an offline mode (ST_REQ_06). Expands the SW_SPEC_01
SW_SPEC_05	**Algorithm for location map suggestion:** Implement an algorithm that suggests a location mapping of devices that are going to be put into the die tray. The algorithm will receive as input a set of device dimensions (x,y) and a positioning strategy. Then based on the health index table, will return a set of locations where these devices will be put. This specification is linked with ST_REQ_04.
SW_SPEC_06	**Provide an API.** The smart tagging application will expose its functionality through an API. This API will be used from ALPES system (ST_REQ_03). Furthermore, a User interface should be implemented (ST_REQ_05). Finally, as a part of the API is the communication with ficonTEC machinery (ST_REQ_07)

3 RFID theory and Hardware specifications

This chapter depicts a quick view of the state-of-the-art smart tagging technologies along with the equipment selection process for the proposed solution.

3.1 Theoretical background

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify, and track tags attached to objects. An RFID system consists of the following components: a radio transponder, a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to track inventory goods. A RFID system uses tags, or labels attached to the objects to be identified and two-way radio transmitter-receivers, called interrogators or readers, are used to send a signal to the tag and read its response. There are many different classifications on RFID systems based on specific characteristics. The main classifications can be based on:

- Operational frequency
- Type of communication between tag and reader

3.1.1 Frequency based classification

Based on operational frequency, RFID are distinguished to low frequency (LF), high frequency (HF) and ultra-high frequency (UHF).

Low frequency (LF)

These RFID systems operate in the 30 KHz to 300 KHz range and have a read range of up to 10 cm. While they have a shorter read range and slower data read rate than other technologies, they perform better in the presence of metal or liquids (which can interfere with other types of RFID tag transmissions). Common standards for LF RFID include ISO 14223 and ISO/IEC 18000-2. LF tags are used in access control, livestock tracking, and other applications where a short-read range is acceptable.

High Frequency (HF)

HF systems operate in the 3 MHz to 30 MHz range and provide reading distances of 10 cm to 1 m. Common applications include electronic ticketing and payment and data transfer. Near Field Communication (NFC) technology is based on HF RFID and has been used for payment cards and hotel key card applications. Other types of smart card and proximity card payment and security systems also use HF technology. Standards include ISO 15693, ECMA-340, ISO/IEC 18092 (for NFC), ISO/IEC 14443A and ISO/IEC 14443 (for MIFARE and other smart card solutions).

Ultra-High Frequency (UHF)

These systems have a frequency range between 300 MHz and 3 GHz, offer read ranges up to 12 m and have **faster data transfer rates**. They are more sensitive to interference from metals, liquids, and electromagnetic signals, but new design innovations have helped mitigate some of these problems. UHF tags are much cheaper to manufacture, and as such are commonly used in retail inventory tracking, pharmaceutical anti-counterfeiting, and other applications where large volumes of tags are required. The EPC global Gen2/ISO 18000-6C standard is a well-known global standard for item-level tracking applications.

3.1.2 Power source-based classification

This classification of tags is based on how the tag communicates with the reader and especially if they have their own power supply or not. There are two types: a) passive and b) active. A hybrid solution can be defined as a third type.

Active RFID

Active RFID tags have their own transmitter and power source (usually a battery) onboard the tag. These are mostly UHF solutions and read ranges can extend up to 100 m in some instances. Active tags are usually larger and more expensive than their passive counterparts and are used to track large assets (like cargo containers, vehicles, and machines). Active RFID tags are also often equipped with sensors that measure and transmit temperature, humidity, light, and shock/vibration data for the objects they are attached to.

There are two types of active tags. Transponders only "wake up" and transmit data when they receive a radio signal from a reader. For example, a transponder attached to a vehicle in a toll payment or checkpoint control location would only be active when passing through a particular gate. This helps conserve battery life. Beacons, on the other hand, emit a signal at a pre-set interval. This type of active tag is used in real-time location systems (RTLS) for tracking anything from wheelchairs at a hospital to large cargo containers at a shipping dock.

Passive RFID

In passive RFID solutions, the reader and reader antenna send a signal to the tag, and that signal is used to power on the tag and reflect energy back to the reader. There are passive LF, HF, and UHF systems. Read ranges are shorter than with active tags and are limited by the power of the radio signal reflected back to the reader (commonly referred to as tag backscatter).

Passive tags are usually smaller, less expensive and more flexible than active tags. This means they can be attached or even embedded on a wider variety of objects. Passive UHF tags are commonly used for item-level tracking of consumer goods and pharmaceuticals, for example.

Battery-Assisted Passive

A third, hybrid type of RFID tag has also emerged. BAP systems, or semi-passive RFID systems, incorporate a power source into a passive tag configuration. The power source helps ensure that all of the captured energy from the reader can be used to reflect the signal, which improves read distance and data transfer rates. Unlike active RFID transponders, BAP tags do not have their own transmitters.

3.1.3 Tags

An RFID tag consists of three different components: an RFID chip comprised of an integrated circuit (IC), an antenna and an underlying layer/base (known as the substrate).

Integrated circuit

An IC, also known as a microchip, is designed and manufactured by a semiconductor manufacturer. Contained within the IC is a logic unit in charge of making decisions and allocating memory to store data. How the IC obtains power depends on the type of tag; in an active tag, it comes from a battery on the tag, while in a passive tag, it is obtained from the interrogating electromagnetic signal generated by the reader. The memory on the IC may be divided into different blocks called banks. Tags use EEPROM memory type, which does not require continuous power to store data and thus can retain data for a long period without the need for power. Data type stored depends on the protocol used. The IC may store tag ID, object identifier, password and error detection codes. An example of tag chips characteristic is shown in Figure 5.

ICs are created on a large semiconductor wafer, which can contain 40,000 ICs. They are manufactured in state-of-the-art clean room facilities and individually tested to ensure reliability. IC designs are headed towards becoming smaller, which reduces the cost and power needed to function.

Impinj Monza 4 Tag Chip Series

		Impinj M4D	Impinj M4E	Impinj M4i	Impinj M4QT
SPECIFICATIONS	Air interface protocol	RAIN RFID / ISO 18000-63 and EPCglobal Gen2v2 compliant			
	EPC memory	128 bits	496 bits	256 bits	128 bits
	User memory	32 bits	128 bits	480 bits	512 bits
	Read sensitivity (dBm)	-19.5	-19.5	-19.5	-19.5
	Write sensitivity (dBm)	-16.7	-16.7	-16.7	-16.7
	Die size (μm)	590 x 590	590 x 590	590 x 590	590 x 590
FEATURES	Impinj TagFocus™ read redundancy prevention	✓	✓	✓	✓
	Impinj FastID™ high-speed reading	✓		✓	✓
	Impinj True3D™ omni-directional reads	✓	✓	✓	✓
	Tag data protection features	Access/Kill	Access/Kill	Access/Kill	QT technology Access/Kill Short-range mode
	Packaged parts		✓		✓

Impinj product performance is based on Impinj's modeling and test data, actual results may vary.

Figure 5: Characteristics of Impinj tag chips (Monza Series 4)

Tag Antennas

The largest connected part of the IC of a tag is called the antenna. This component receives signals from the interrogator and depending on the tag type, will either transmit or reflect the received signal. For active, it transmits the signals, and for semi-passive and passive tags, it reflects the signals. Furthermore, for passive tags, the antenna also collects power from the radio waves and supplies it to the IC.

The shape of the antenna is determined by the tag's operation frequency. Variations in the design of the antenna allow tags to have different properties and behaviors. The antennas can be shaped like a spiral coil, a single dipole, two dipoles perpendicular to the other, or a folded dipole. Many variations of these basic types exist and are chosen depending on the specific requirements of its users and the abilities of the designer. The antenna is designed for a specific frequency and is later tuned to be appropriate for properties of the materials to be tagged. This frequency determines the effective antenna length, but the actual length is usually shortened by design.

The antennas are usually made from thin metal strips of copper, aluminum, or silver. They are placed on the substrate via one of three methods: copper etching, foil stamping, and screen-printing. Screen-printing is the fastest and cheapest method but creates the least efficient antennas

RFID Tag Substrates

This is the layer that holds all the other components together. A substrate is usually made from a flexible material such as thin plastic but can also be made from a rigid material. Most passive tags use substrates of flexible material. The material is selected to withstand various expected environmental conditions that the tag is to encounter in its lifecycle. Some materials include polymers, PVC (polyvinyl chloride), polyesters, styrene, and paper. The material used must provide a smooth printing surface for antenna layout, durability, and stability under various conditions, dissipation of static build-up, and mechanical protection for the components and their interconnections. The substrate is designed to withstand conditions such as sunlight, chemicals, heat, moisture, vibration, impact and corrosion.

The substrate material may affect the design frequency of the antenna, which must be considered during the proper tuning of the antenna. Adhesive material designed to withstand the appropriate environmental conditions is used on one side of the substrate to attach the tag to an object. A protective overlay made from materials such as PVC lamination, epoxy resin or adhesive paper is sometimes added to provide additional protection.

3.2 Specifications on tagging system

The requirements from the previous chapter (paragraph 2.3) generate a set of specifications for our hardware selection. These specifications are related mainly to the characteristics and the operation of tags. The non-mandatory requirements deal with reader selection.

HW_SPEC_01	**RFID tag with memory**. Based on ST_REQ_01, the RFID chip inside the tag should provide sufficient user memory to store a health index table for each die tray cell.
HW_SPEC_02	**RFID tag dimensions.** This specification is acquired from ST_REQ_01 and ST_REQ_02 in combination with RFID tag antenna characteristics. The bigger the antenna the bigger the distance where the tag can be read from the reader. Finally, the tag should fit with the die tray dimensions.
HW_SPEC_03	**Installation of multiple antennas.** The requirement ST_REQ_08 can be met with two alternative options. The first option is to use a separate reader and antenna where a reader can support multiple antennas. Each antenna can be used to support the execution of a scenario. The second option is to have an antenna that can read multiple tag (sequent or parallel)
HW_SPEC_04	**The reader should provide an API.** This specification comes from the non-functional requirement that the system should be detached from hardware selection. Of course, a specific hardware that satisfies the specifications above will be selected and tested.

3.3 Hardware prerequisite selection

3.3.1 RFID tag

Regarding the specifications above, the RFID tag should include a chip with sufficient user memory in order to store information related to the die tray life cycle. The most consuming variable is the health index table. Based on the feedback from ALPES each cell can be used approximately 10 times at most before it wears down. Also in our use case, a grid-based die tray will be selected with capacity of 10X10 cells. All other variables consume less than 100 bits memory space in total. Most FRID chips provide memory capacity in a range between 32 to 512 bits. In some cases, 1024 memory chips can be acquired. In our use case an RFID tag with 512 bits memory chip will be used (Impinj MQ4T). For die trays with a capacity larger than 100 cells a RFID tag with larger memory can be applied.

Another limitation to the selection of RFID tag is the dimensions. The dimension of the tag is related with the RFID antenna dimensions. Large antenna means that can interact with the reader in larger distance. In our use case the distance between antenna (reader) and RFID tag will be less than 20 cm. This limitation is due to the dimensions of the selected die trays. Typical dimension for a die tray ranges from about 7 cm x 7 cm to 9 cm x 12 cm. An example of a selected RFID tag is depicted to the Figure 6.

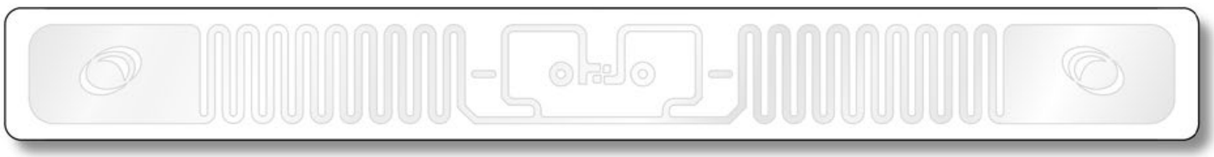


Figure 6: Antenna of an RFID tag.

4 System Architecture and implementation

4.1 High level architecture

The smart tagging system consists of two separate layers: the hardware layer and the software layer. The hardware layer is a typical RFID system that "tags" and recognizes each monitored die tray. The technology to this layer is well defined and there are many solutions for each hardware component. Depending on the selected solution, an API or SDK is offered for the interfacing between hardware and software parts that supports the basic read/write functionalities. The software layer implements the business logic of the smart tagging application. We detached these two layers in our architecture, in order to make our proposed solution more abstract and hardware agnostic. This approach can provide the flexibility to expand the smart tagging application with different hardware, just by implementing a new read/write controller.

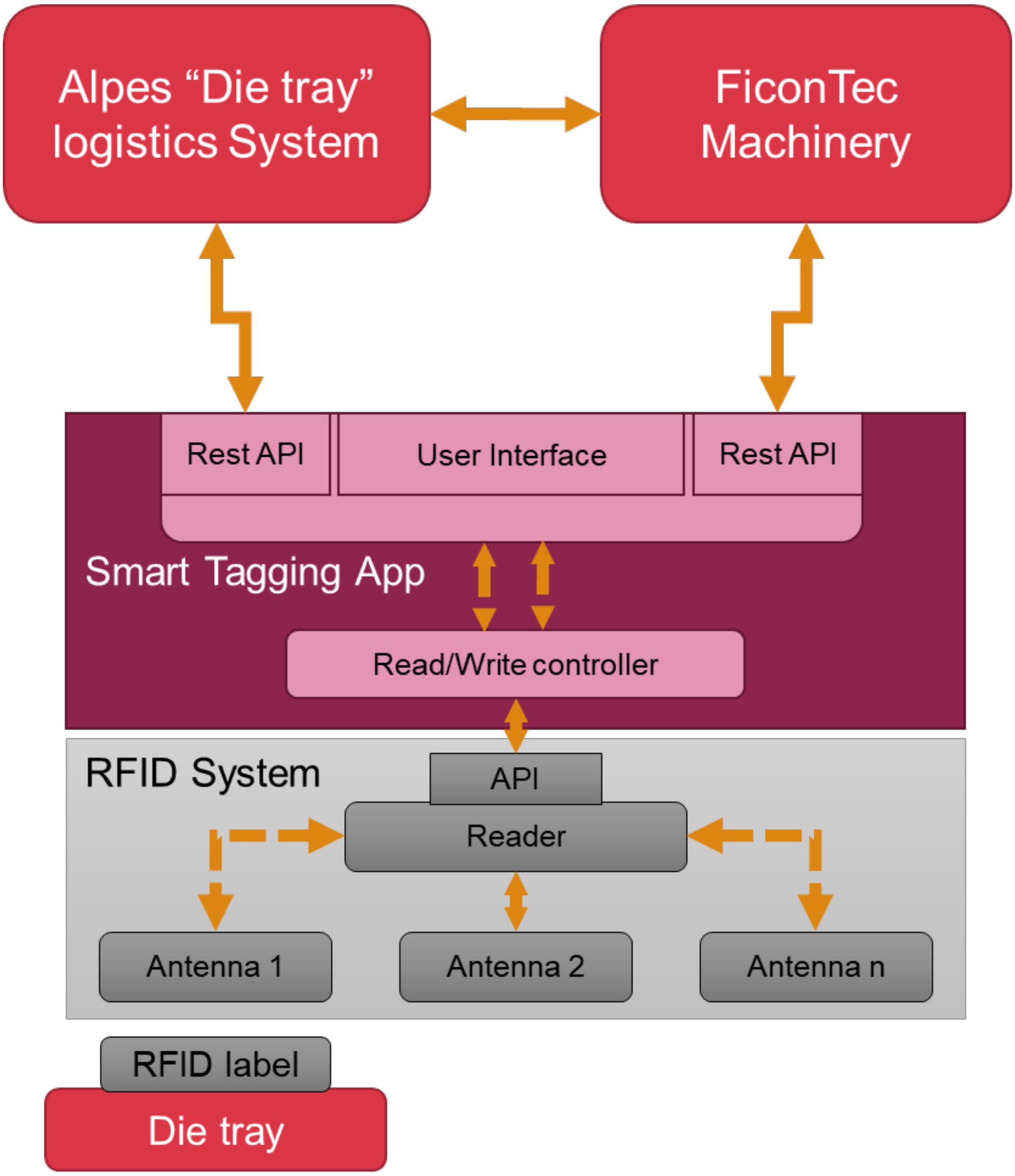


Figure 7: Smart tagging high level architecture

4.1.1 Components

The Smart Tagging System is divided into the following components:

- Software part (Smart Tagging App)
    - Read/Write controller. This component is responsible for the communication with the RFID system. Converts all requests from the upper layer to simple read/write commands and communicate with the reader SDK or API.
    - Rest API. Exposes the core functionalities of the Smart Tagging System to external systems such as ALPES System or the ficonTEC machine.
    - User interface. An operator can interact with the Smart Tagging System
- Hardware part (RFID system)
    - RFID tag. For each die tray a new RFID tag will be initialized and attached on it. Each tag will contain a unique id that it will finally define the die tray.
    - RFID antenna. It is responsible for the communication between RFID tag and smart tagging application. Main criteria for antenna selection are the range and the ability to read multiple tags. In our use case we have a range less than 30 cm. In order to support multiple scenarios (empty and fill) multiple antennas can be used.
    - RFID reader: This is the main hardware for the smart tagging system. Depends on the vendor or hardware solution, reader and antenna can be different equipment or a combo solution. In our case we selected separate solutions in order to connect multiple antennas and support combination of the defined scenarios.

4.2 Functionality

4.2.1 Functionality provided by the Smart Tagging application

Full specifications of the provided functionality are depicted to the Annexes. In paragraph 7.1 “Rest API”, rest API functionalities are provided and in paragraph 7.2 “User Interface” the user interfaces are depicted. To this paragraph we describe the basic functionality of the Smart Tagging System:

- **HTTP GET /DieTrays/Get (getDieTrays):**
  This operation returns all available die trays that are in the reader’s range. It is used from the operator via the user interface in order to select the die tray. This method is provided through the user interface.
- **HTTP GET /DieTrays/Get{DieTrayId} (getDieTray):**
  This operation returns the health status data for the selected die try. Health data are the expiration date, the type of die tray and the health index table for containing cells. This method is provided through the user interface.
- **HTTP POST /DieTrays/Init (initDieTray):**
  Initialize a new die tray. For the initialization of a new tag that relates to a die tray we should define a new dieTrayId, the type of die tray, the max number of usages for each cell and finally, the expiration date for this die tray. This method is provided through the user interface.
- **HTTP POST /DieTrays/SetCellHealth/{DieTrayId}:**
  This method is used for manually updating the health index for a cell of the selected die tray. This is used in case that a specific cell of the die tray is damaged (usually during the transportation). This method is provided through the user interface.
- **HTTP GET /reader/getStatus (readReaderStatus):**
  This method is used to retrieve the reader status. It is useful for checking the operation of the reader and antenna operation status. This method is provided through the user interface.
- **HTTP POST /DieTray/SuggestMapping/{DieTrayId}**
  This method is used to retrieve a suggested mapping for a set of devices that are going to be stored to the selected die tray. As input a list of devices dimensions (x,y), and the position strategy is defined. The output is a list of locations for each device. If the die tray cannot store all devices, the method returns the positioning only for these devices that can be stored.
- **HTTP POST /DieTray/UpdateHealthIndex/{ DieTrayId }**

This method updates the health index for a position inside the selected die tray. If the "onTheFly" status is "true", then the method decreases the health index of the specified position. Otherwise, the method stores this position as an update "job" to commit it on a later time.

- **HTTP POST /DieTray/CommitUpdate/{DieTrayId}**
  This method commits all update jobs for the selected die tray. The update "jobs" were defined with the previous method and marked as offline update.
- **HTTP POST /DieTray/RollBack/{DieTrayId}:**
  This method cancels all update "jobs" for the selected die tray.

4.2.2 Location suggestion and Strategies

A challenging part of the Smart Tagging App is the devices' location selection. The algorithm tries to put devices to the cells that have the higher health index. With this approach, we achieve a uniform decrease of die tray's health status and maximize its usage. Based on use case demands, another parameter should be taken under consideration for the final location selection: the location strategy. According to the location strategy, a set of parameters can be defined that affect the location selection. More specifically, for our use case we defined a location strategy that we call minBordersStrategy{range=1}. According to this strategy, when a device is put on a range of cells inside the die tray, at least a range of a single cell around the device should be kept empty. The range of cells where the device is placed will decrease its health index by a unit's cost. All empty cells around the device location will retain the same health index. For an example, let's suppose that we have 5 devices that should be mounted into the die tray: {D1(2,3), D2(1,2), D3(1,2), D4(1,3), D5(2,1)}. Also, the health index table for the selected die tray is depicted in Figure 8.

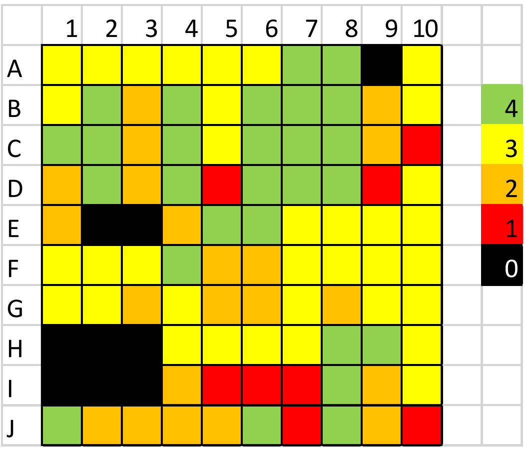


Figure 8: Health index table for 5 usages

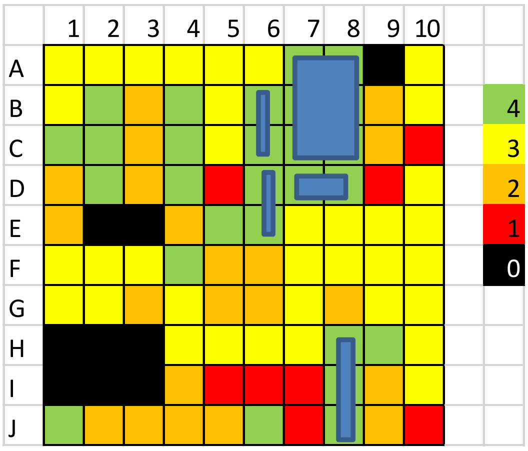


Figure 9: Location without location strategy

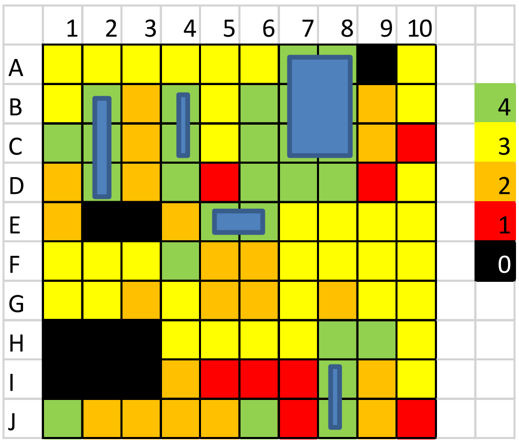


Figure 10: Location with location strategy minBordersStrategy{range=1}

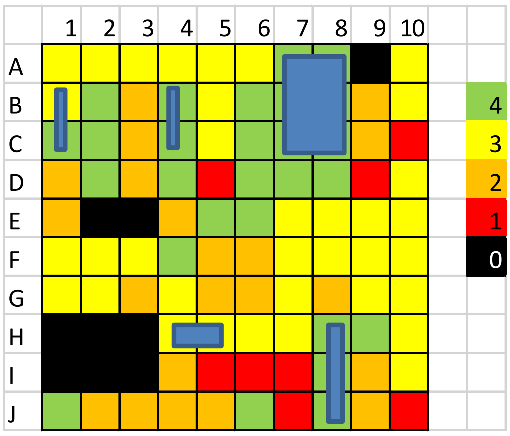


Figure 11: Location with location strategy minBordersStrategy{range=2}

Based on our concept, we should use cells with the higher health index (4 if possible) and leave empty, the cells that have lower (1 or 2). Any cell with health index equal to 0, cannot be used. In our example there are many cells with the higher health index (4) and without using a location strategy, there are many possible mappings. One of the possible location mappings for these devices is: {A7C9, B6C6, D6E6, H8J8, D7D8}, which is depicted in Figure 9. If, in our example, we use the defined location strategy, the mapping may differ from the previous one, since the limitations that were introduced from the location strategy are prioritized, compared to the best available health indexes. Figure 10 {A7C9, B4C4, I8J8, B2D2, E5E6} and Figure 11 {A7C9, B4C4, B1C1, H8J8, H4H5} depict a possible mapping when applying the location strategy with different value (range=1 and range =2).

4.2.3 Interfacing with other systems

As mentioned in the executive summary, the scope of the proposed system is from one hand to convert a product to “smart” and – on the other hand – to automatically setting-up the process chain parameters. As a smart product the die trays were selected. The process that was chosen to be automated is the packaging of optoelectronic devices and the transportation between the different production stages. The smart tagging system in cooperation with the following systems that already involved to the current situation achieve the goal of automatic setting-up. The systems that are involved into this process are:

- ALPES “Die tray” logistic system. The scope of this system is to store information of all jobs/mission that are assigned to each die tray. ALPES system stores the location of each device inside the die tray and follows the life cycle for each individual device. On the other hand, the ALPES system does not have any information that is related with health status of each die tray.
- ficonTEC machine: The ficonTEC machine as part of the whole iQonic solution is involved, since it fills with devices the die trays and uses these devices to assemble the final optoelectronic product. The machinery setting-up process requires the full location mapping of each device inside the die tray. Since this information is available, the machinery can unmount a specific device and use it to the production assembly process. The location mapping can be directly extracted from ALPES system. The Smart Tagging Application provides to the machinery the id of the selected die tray, in order to extract the location mapping.

4.3 System use-cases description

In this paragraph, the evaluation of system functionality is performed based on the scenarios defined previously in paragraph 2.2 (Scenario definition).

4.3.1 Initialize a die tray

This scenario (Figure 12) is used to define a new die tray to the logistics system and is controlled by an operator through the provided user interface. Following, we depict the steps for this scenario.

1. Position the die tray near to the antenna of RFID system. Of course, an RFID tag should be sticked in each die tray. The RFID tag will store all data that are related to the die tray.
2. Through UI define all necessary variables regarding the die tray initialization. Before this step, the user prompts to select the correct die tray from a list of die trays that are in the range of the RFID system. This pre-step is used in order to avoid the redefinition of a die tray that incidentally is on the

range of the antenna. The variables that the operator should define are: the type of die tray, the unique id, the max number of usages for each cell and the die tray expiration tray.

3. The Smart Tagging App triggers RFID system to initialize RFID tag. After the operator submits the request, the Smart Tagging App, through the read/write controller, sends a write request to the reader controller API.
4. The reader, through the antenna, writes to the RFID tag the data defined above.
5. Since the write status is successful, the Smart Tagging App sends to the ALPES system the details of the new die tray.
6. Done.

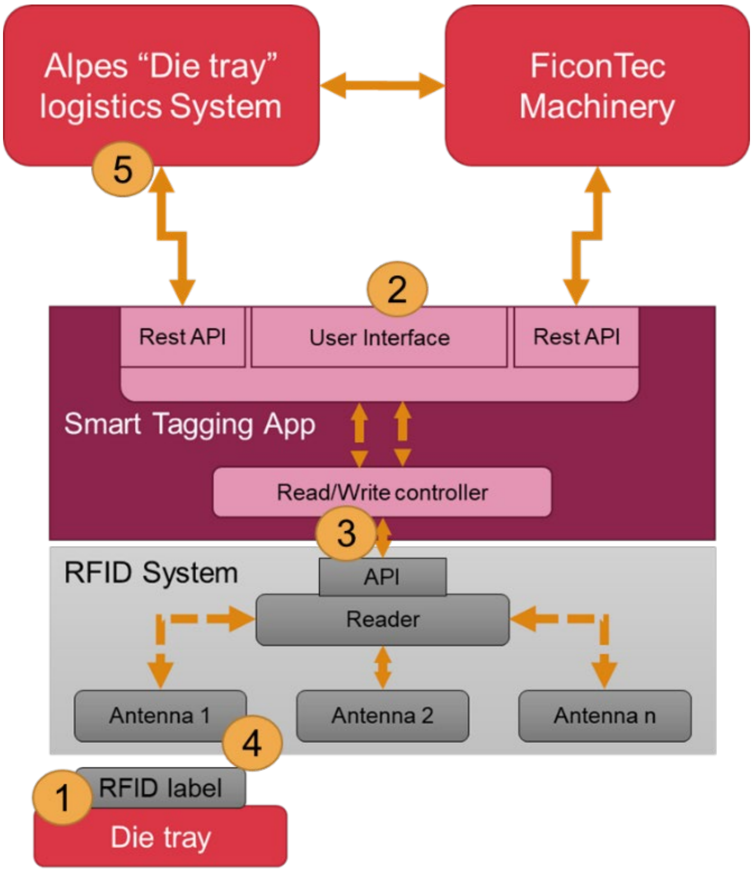


Figure 12: Initialize a die tray

4.3.2 Read a die tray

This scenario is useful to check if a die tray is suitable for a new transportation (Figure 13). It provides a graphical status of each die tray. Following, we depict the steps for this scenario:

1. Put die tray to the RF-ID controller inside the antenna range.
2. From the UI, the operator can see a list of all die trays that are in the antenna range. From this list choose which die tray to be used.
3. Since a die tray is selected, the Smart Tagging App, through read/write controller, requests the health index details.
4. The reader reads data from the tag and returns these data to the smart Tagging App.
5. The UI depicts the health status and population information of the die tray.
6. Done.

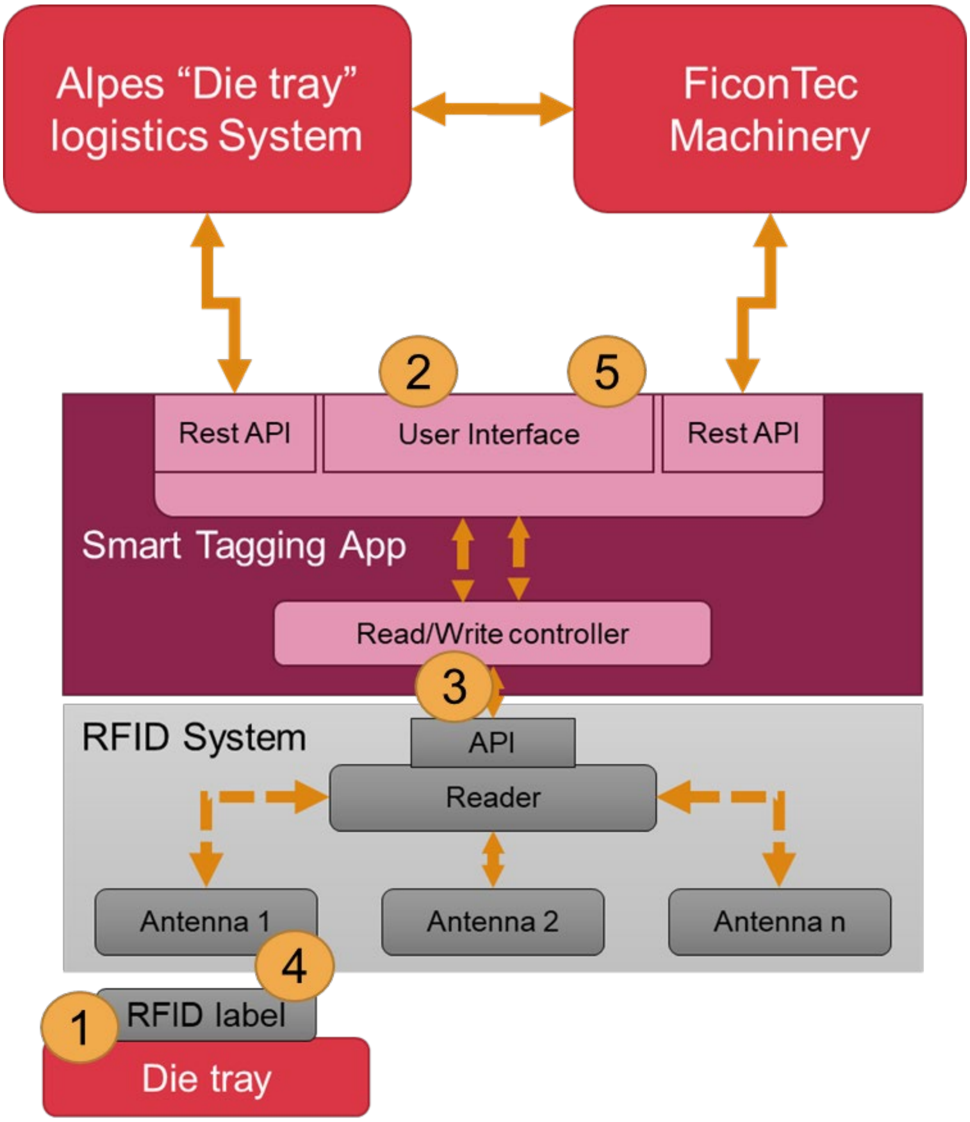


Figure 13: Read a die tray

4.3.3 Mount a die tray with devices

This scenario is used every time a die tray is used to carry and transport optoelectronic devices between different production stages (Figure 14). Following, we depict the steps for this scenario:

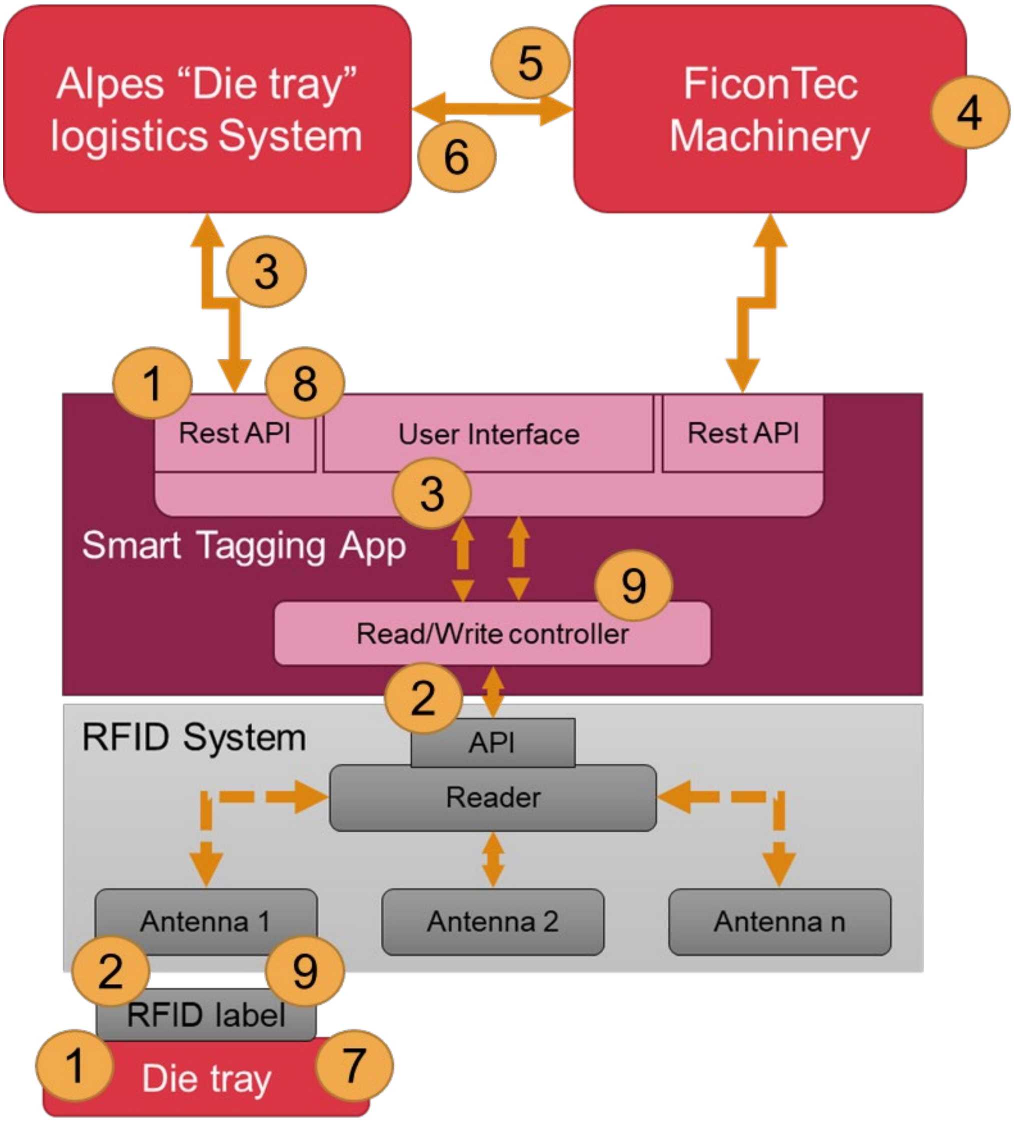


Figure 14: Fill a die tray with devices

1. ALPES system requests a Location topology inside die tray for a list of devices. The die tray must be inside the range of the antenna. The ALPES system then sends a "Fill request" to the Smart tagging App.
2. The Smart tagging App requests from RFID system – through reader API – health index data for the selected die tray. The reader gets the data from tag and returns them back to the Smart Tagging App
3. The Smart Tagging App, based on the current health status table of die tray and defined location strategy, calculates and proposes the best fitting location for each device. Then, returns to the ALPES system a list of locations.
4. Since the location mapping is done, the die tray is placed to the machinery that will fill it with the devices.
5. ALPES system sends the Location Matrix to the FICONTEC machine.
6. Each time the machinery puts a device, ALPES system updates to the Smart Tagging App the device location inside the die tray. Furthermore, the ALPES system marks if the update status is on the fly or not.
7. When all devices are mounted onto the die tray, the die tray gets placed inside the antenna range.
8. ALPES system triggers the commission of the changes into health status. Two options are available: a) commit changes and b) rollback. If jobs that were defined in step 6 are done on the fly, means that the die tray is inside the range of antenna and update was done on the fly. In this case this step can be skipped and there is no rollback option.

9. The Smart tagging app updates the health index table. Again, the update is done through reader API. If the update is on the fly this step follows the step 6.
10. Done.

4.3.4 Empty a die tray

This scenario depicts the process where a die tray arrives to its destination and provides the stored devices to the next production step (Figure 15). This scenario can be executed multiple times since all stored devices are used. Following, we depict the steps for this scenario:

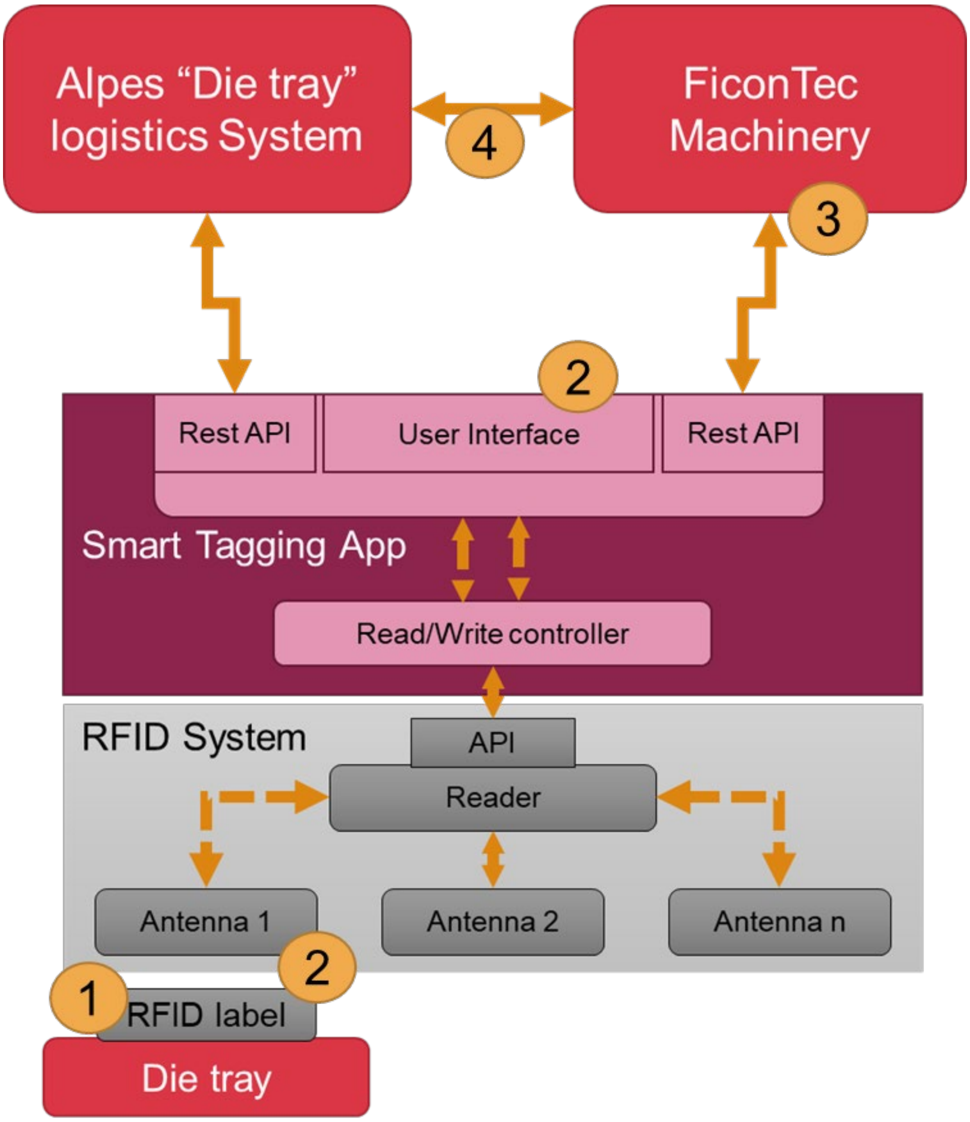


Figure 15: Empty a die tray

1. Place the die tray to near the RFID antenna.
2. Execute the scenario "read the die tray".
3. The smart Tagging App sends the id of the selected die tray to the machine.
4. The machine retrieves the Devices Location from ALPES system based on the die tray id and gets the devices that will be used in the next production stage.
5. Done.

4.3.5 Manual update health status

The final scenario is the manual update of health index (Figure 16). This is used when the die tray is partially or totally damaged. Following, we depict the steps for this scenario:

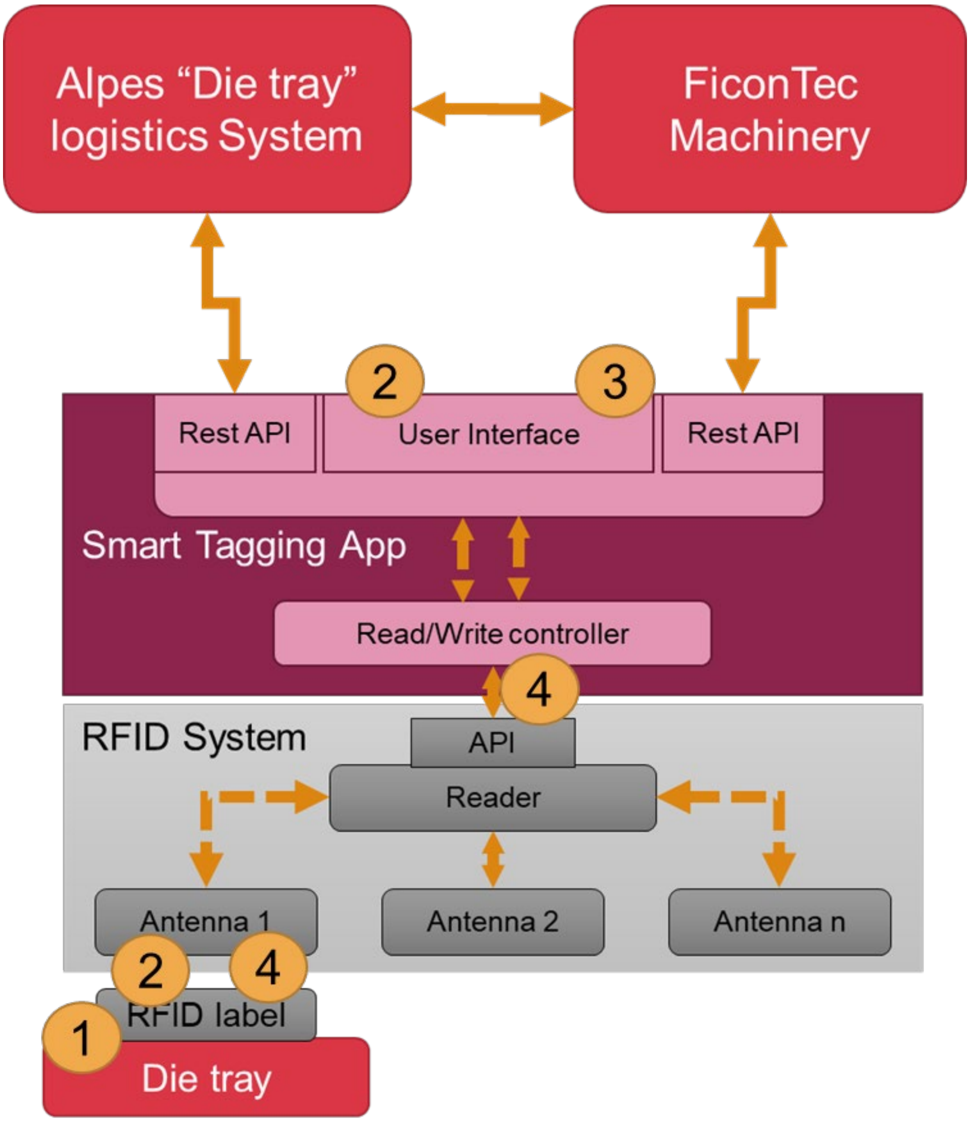


Figure 16: Manually update of health index table

1. Place the die tray near the RFID antenna.
2. Execute the scenario “read the die tray”.
3. Operator updates the health index table via the user interface. Every cell or area that is damaged is flagged as “bad”.
4. The Smart Tagging App updates the health index table. The update command is sent to the reader via the read/write controller and finally, the data are stored to the tag
5. Done.

4.4 Implementation

The Smart Tagging System consists of the Smart Tagging App, which is the software aspect of the system and the RFID system containing the hardware aspects of the overall system. The hardware selection presented in this paragraph was performed based on the hardware prerequisites described in the previous chapters.

4.4.1 Smart Tagging App Implementation

The implementation of the software aspect of the Smart Tagging System was developed in .NET core and is Platform independent. Smart Tagging App is installed in a Linux based Daedalus solution that relates to tasks 4.1 and 3.4.

4.4.2 RFID System Implementation

For the hardware implementation of the RFID System, two main devices must be selected based on the requirements and specifications of chapter 2, the Reader and the Antenna.

The selected Reader is the Impinj Speedway R420, that has 4 ports for connecting antennas and can be expanded up to 32. The main reasons behind this choice are the fact that Impinj is considered a leader in RFID technology, provide and API/SDK and can support multiple antennas since more than one are needed in our case. The first antenna could be positioned at the area of the application, so that it would automatically collect the tag's ID and process this information in SENSAP's application accordingly. The second or more antennas could be then positioned in the cabinet where the die trays are stored for example, so that we can have the option to provide a live inventory and asset management possibility in the future.

More specifically, the R420 Impinj Speedway Reader involves the RAIN RFID Air interface protocol and is ISO 18000-63 and EPC global Gen2v2 compliant. Contains, as mentioned, 4 antenna ports and has a maximum read zone of 32. Its maximum read rate is 1,100 per second while the maximum transmit power reaches 32.5 dBm. It can receive sensitivity of -84 dBm, has a 0.4 GHz processor speed with 256 MB Random-access memory and 32 MB Custom-application partition (CAP) size. The key benefits of RAIN RFID readers can be summarized as following:

- Optimal RAIN RFID Performance. They can maintain high read rates regardless of RF noise or interference as the readers leverage built-in automation features to ensure peak performance.
- Versatile and Customizable. They are supported by a suite of hardware accessories and antennas that deliver deployment flexibility streamlining installation and expansion.
- Powerful Application Development Tools. There is a variety of development libraries and software for purpose built faster application development.
- Enterprise-Class Reliability. They are considered to have industry-best enterprise-class reliability for trouble-free operation and long life.

Regarding the selected antennas for the system, we have defined the following two options. The Impinj MatchBox RAIN RFID Antenna and the Circular Polarized UHF Antenna SlimLine – A5010.

The Impinj Matchbox antenna is a very small RAIN RFID antenna suited for embedded applications needing strong performance in a tight read zone. This antenna is part of a custom solution built with a Speedway reader and is ideal for tight-proximity spaces that need exceptional control in a small zone. The antenna is unobtrusive and can monitor items within cabinets or other small enclosures. More specifically it has a frequency range of 865 to 956 MHz broadband for use in all regions and -20 dBi far-field gain. An impedance of 50 ohms and offers linear polarization (parallel to short axis). It has 2 mounting options and an IP rating of IP54 for indoor use only. It weights 0.02 kgs (0.04 lbs) and has SMA female connector that requires accessory cable to connect to the reader's RP-TNC connector. The key benefits of RAIN RFID readers are the compact design with dimensions of 7.3 x 3.3 x 1.1 cm (with 20 cm (8 in) pigtail) with diminutive design that fits behind displays to provide protection for high-value items, its bi-directional, short-range read zone coverage and its broadband support for multi-region operation and optimal performance globally.

The SlimLine – A5010 Circular Polarized UHF Antenna is an award-winning UHF antenna that provides fast and accurate tracking with high levels of performance and durability. It has an IP67 environmental rating for permanent outdoor use meeting high standards of quality and robustness. More specifically, it consists of a UV-Resistant ABS random material and has a frequency range of 865-868 MHz / 902-928 MHz, has RHCP (Right Hand Circular Polarized) polarization and 8.5 dBiC typical far-field gain with 68 degrees far-field 3 dB beamwidth in both planes. It has a nominal impedance of 50 Ω and 10 K Ω resistance antenna detection. Furthermore, this antenna offers 1.3 typical VSWR with -20dB front to back ratio and DC grounded anti-static protection. Its maximum input power is 3 W, it offers 2 mounting options and a SMA female side connector. Finally, its operating temperature can be -20° to +55°C and its storage temperature -30° to +65°C.

5 Conclusions and Next Actions

The Smart Tagging System was implemented and tested in laboratory as a prototype. All scenarios tested and evaluating the Smart Tagging Application functionalities. As next action, it will be integrated with other systems and tested to the use case environment. These actions will be performed under WP7 and task 7.1. Briefly, the evaluation will be focused on the following topics a) on site equipment installation b) software deployment on Daedalus infrastructure regarding to Task 4.1 and task T3.4 c) UI evaluation from operators d) integration tests with ALPES system e) integration tests with ficonTEC machine and f) fine tune of the suggestion algorithm to production conditions.

6 List of Figures

7 Annexes

7.1 Rest API

Title	**Get all die trays that in antenna range**
Description	Get all die trays that in antenna range
URL	DieTrays/Get
Method	GET
Request headers **Content-Type:** application/json	**Response headers** **Content-Type**: application/json

Request parameters	**Response body**
	{
	"dieTrays":
	[
	"id":"string",
	]
	}

Title	**Get health data for the selected die tray**
Description	Get health data for the selected die tray
URL	DieTrays/Get/{DieTrayId}
Method	GET
Request headers **Content-Type:** application/json	**Response headers** **Content-Type**: application/json

Request parameters
 	
DieTrayId: the id of the selected die tray
	Response body
	{
	"expirationDate": "date",
	"dieTrayType":"string",
	"maxUsageForEachCell": integer,
	"healthIndexTable":
	[
	{
	"index": string
	"healthStatus":integer
	},
	]
	}

<table>
<tr><td>Title</td><td colspan="2">Initialize a new Die Tray</td></tr>
<tr><td>Description</td><td colspan="2">Initialize a new Die Tray</td></tr>
<tr><td>URL</td><td colspan="2">/DieTrays/Init</td></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td colspan="2">Request headers
Content-Type: application/json</td><td>Response headers
Content-Type: application/json</td></tr>
<tr><td colspan="2">Request body
{
“factoryDefaultId”: “string”,
“dieTrayId”:”string”,
“dieTrayType”:”string”,
“maxUsageForEachCell”: integer,
“expirationDate”: “date”,
}</td><td>Response body
{
“success”: “Boolean”,
“dieTrayId”:”string”
}</td></tr>
</table>

<table>
<tr><td>Title</td><td colspan="2">Suggest location mapping for a list of devices to the selected die tray</td></tr>
<tr><td>Description</td><td colspan="2">Suggest location mapping for a list of devices to the selected die tray</td></tr>
<tr><td>URL</td><td colspan="2">/DieTray/SuggestMapping/{DieTrayId}</td></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td colspan="2">Request headers
Content-Type: application/json</td><td>Response headers
Content-Type: application/json</td></tr>
<tr><td colspan="2">Request body
{
"devices":
[
{
"Index": string,
"size_x_mm": float,
"size_y_mm": float
}
],
“locationStrategy”:”LocationStrategyType”
}</td><td>Response body
{
“success”: “Boolean”,
"ArrayList":
[
{"Index": string,
"Position": string
}
]
}</td></tr>
</table>

<table>
<tr><th>Title</th><th colspan="2">Update health index for selected die tray</th></tr>
<tr><td>Description</td><td colspan="2">Update health index for selected die tray</td></tr>
<tr><td>URL</td><td colspan="2">DieTray/UpdateHealthIndex/{DieTrayId }</td></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td colspan="2">Request headers
Content-Type: application/json</td><td>Response headers
Content-Type: application/json</td></tr>
<tr><td colspan="2">Request body
{
"Positions":
[
{
"PositionValue":"A3B4"
}
] ,
“onTheFly”:”boolean”
}</td><td>Response body
{
“success”: “Boolean”,
}</td></tr>
</table>

<table>
<tr><th>Title</th><th colspan="2">Commit update changes to the selected die tray</th></tr>
<tr><td>Description</td><td colspan="2">Commit update changes to the selected die tray</td></tr>
<tr><td>URL</td><td colspan="2">/DieTray/CommitUpdate/{DieTrayId}</td></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td colspan="2">Request headers
Content-Type: application/json</td><td>Response headers
Content-Type: application/json</td></tr>
<tr><td colspan="2">Request body</td><td>Response body
{
“success”: “Boolean”,
}</td></tr>
</table>

<table>
<tr><th>Title</th><th colspan="2">Rollback update changes to the selected die tray</th></tr>
<tr><td>Description</td><td colspan="2">Rollback update changes to the selected die tray</td></tr>
<tr><td>URL</td><td colspan="2">/DieTray/ RollBack /{DieTrayId}</td></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td colspan="2">Request headers
Content-Type: application/json</td><td>Response headers
Content-Type: application/json</td></tr>
<tr><td colspan="2">Request body</td><td>Response body
{
“success”: “Boolean”,
}</td></tr>
</table>

7.2 User Interface

7.2.1 Read all die trays are in the antenna range

Read Die Tray

Die Tray Id	
F23456789012345678901234567 9801B	Read Die Tray
F23456789012345678901234567 9801C	Read Die Tray
F23456789012345678901234567 9801A	Read Die Tray

7.2.2 Read health data for a selected die tray

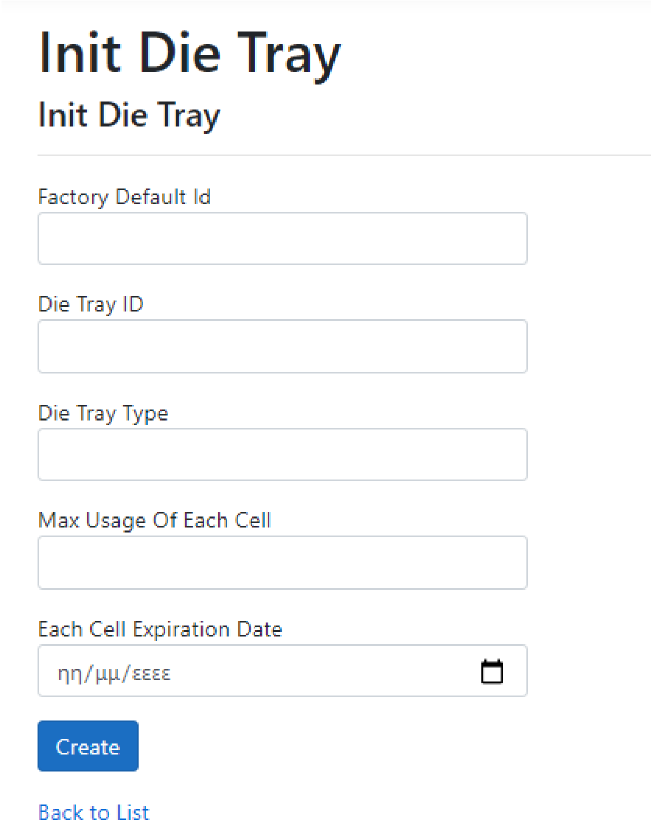
Die Tray

Use Die Tray

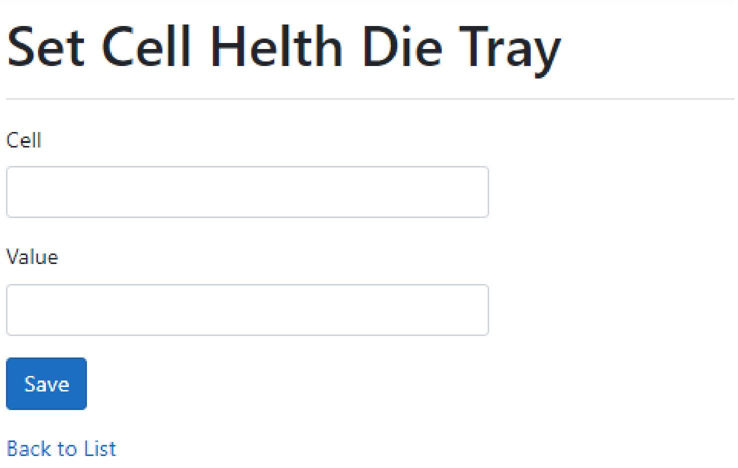
Expiration Date	2021-04-30
Die Tray type	GP-G-ANT-10x10
Max Usage Of Each Cell	5

	A	B	C	D	E	F	G	H	I	J
1	0	4	5	4	0	5	2	1	4	2
2	4	5	4	0	5	2	1	4	2	5
3	5	4	0	5	2	1	4	2	5	0
4	4	0	5	2	1	4	2	5	0	2
5	0	5	2	1	4	2	5	0	2	2
6	5	2	1	4	2	5	0	2	2	2
7	2	1	4	2	5	0	2	2	2	5
8	1	4	2	5	0	2	2	2	5	5
9	4	2	5	0	2	2	2	5	5	1
10	2	5	0	2	2	2	5	5	1	5

7.2.3 Initialize a new die tray



7.2.4 Set health for a cell



7.2.5 Reader Status

Reader Status

ModelName	Speedway R420
ModelNumber	2001002
ReaderModel	SpeedwayR420
FirmwareVersion	6.4.1.240
AntennaCount	4

Reader Features

IsConnected	[x]
IsSingulating	[ ]
Temperature	29

Reader Settings

RMode	AutoSetDenseReaderDeepScan
Smode	DualTarget
Session	2
Antenna1RxSensitivity	-80
Antenna2RxSensitivity	-80
Antenna3RxSensitivity	-80
Antenna4RxSensitivity	-80
Antenna1TxPower	31,5
Antenna2TxPower	31,5
Antenna3TxPower	31,5
Antenna4TxPower	31,5