



Smart Anything Everywhere Initiative

Area 3: Advanced micro-electronics components and Smart System Integration Project: H2020–No 761809



Digital Innovation Hubs boosting European Microelectronics Industry

D3.2: Farm Monitoring and

Control Application Experimentation

Author(s):Th. Zahariadis, P. Athanasoulis, K. Railis, P. KapsalisStatus -Version:FFDelivery Date (DoA):31 May 2018Actual Delivery Date:30 May 2018Distribution - Confidentiality:PublicCode:DIATOMIC_D3.2_SYN_FF-20180530

Abstract:

This manual provides a useful step-by-step guide for the precision agriculture integrated platform.

The integrated platform includes the Synfield platform, the Virtual Fort Knox Cloud platform and the plant-o-meter smartphone application. The Synfield nodes and the plant-o-meter devices collect measurements from the fields in this experiment.

Disclaimer

This document may contain material that is copyright of certain DIATOMIC beneficiaries and may not be reproduced or copied without permission. All DIATOMIC consortium partners have agreed to the full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

The DIATOMIC Consortium is the following:

Participant number	Participant organization name	Short name	Country
01	INTRASOFT International S.A.	INTRA	BE
02	F6S NETWORK LIMITED	F6S	UK
03	BioSense	BIOS	SRB
04	Synelixis Solutions	SYN	EL
05	Instituto Pedro Nunes	IPN	PT
06	Fraunhofer IPA	IPA	DE
07	InoSens	INO	SRB
08	Libelium Comunicaciones Distribuidas SL	LIB	ES
09	FastTrack	FASTT	PT

The information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.



Document Revision History

Date	Issue	Author/Editor/Contributor	Summary of main changes
04.05.2018	V0.1	P. Athanasoulis, K. Railis (SYN)	Initial version of document
15.05.2018	V0.2	P. Athanasoulis, P. Kapsalis (SYN)	Add contributions from IPA and BIOS
17.05.2018	V0.3	Th. Zahariadis, P. Kapsalis (SYN)	SynField Analysis
21.05.2018	V0.4	P. Athanasoulis (SYN)	Finalize document
29.05.2018	V0.5	R. Hermoso (LIB)	Peer Review
30.05.2018	V0.6	P. Athanasoulis (SYN)	Pre-final version
30.05.2018	FF	B. Ipektsidis, D.Rublova (INTRA)	Final version



Table of Contents

1	Int	roduction	10
	1.1	Goal	10
	1.2	Test scenario	11
2	Ма	odules description	12
	2.1	The Plant-O-Metersmartphone application	12
	2.2	SynfieldPlatform	16
	2.3	Virtual Fort Knox Cloud Platform	53
3	Ste	p by step implementation of test scenario for Application Experiment: Agrifood	77
	3.1	A general overview	77
4	An	nex	80
	4.1	Nitrogen rate experiment in maize	80
	4.2	Smart irrigation experiment	82



List of Figures

Figure 1: Agro Application Experiment	11
Figure 2: Operating principle of the Plant-O-Meter	12
Figure 3: Plant-O-Meter (left) and specifically designed smartphone application (right)	12
Figure 4: Main menu of the Plant-O-Meter application	13
Figure 5: Point mode of measurement at an arbitrary location	13
Figure 6: Head node homocentric circles mode of measurement	14
Figure 7: Continuous mode of measurements	15
Figure 8: SynField ecosystem	16
Figure 9: SynField Head Node (left) and Peripheral Node (right)	16
Figure 10: Typical SynField installation	17
Figure 11: Registration in the Synfield platform	17
Figure 12: History of user subscriptions	18
Figure 13: AvailableSMS and remaining API calls	18
Figure 14: Selected SynControl Android app screen	19
Figure 15: Add a new field in your account	19
Figure 16: Register in the platform a new field	20
Figure 17: Manage your fields	20
Figure 18: Register a Synfield head node	21
Figure 19: Step 1- Type the basic details of the Synfield head node	21
Figure 20: Step 2 - Modify the head node's coordinates	22
Figure 21: Register a peripheral node as part a Synfield network	22
Figure 22: Manage the nodes in similar way	23
Figure 23: Add a new sensor in the head node	24
Figure 24: Follow the wizard to complete the sensor's registration	25
Figure 25: Manage the registered sensors	25
Figure 26: Add a new actuator in a Synfield node	26
Figure 27: Follow the wizard to register a new actuator	26
Figure 28: View the registered actuators	27
Figure 29: Add a new external device (e.g. Plant-O-Meter)	27
Figure 30: Register the Plant-O-Meter in Synfield platform	28
Figure 31: Managethe registered devices	28
Figure 32:Current status of sensing services in Synfield node	29
Figure 33: Select sensing services to be visualized	29
Figure 34: Visualize air temperature and relative humidity in one plot	30

diatomic

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation

Figure 25: Affected crop types by specific disease	
Figure 35. Affected crop types by specific disease	30
Figure 36: Diseases Repository	31
Figure 37: Initialising disease prediction in a field	31
Figure 38: Setting disease prediction details	32
Figure 39: Basic step of irrigation rule	32
Figure 40: Define the metrics, necessary for the irrigation rule	33
Figure 41: Define the soil profile	33
Figure 42: Associate one or more actuators with the irrigation rule and define the irrigation plan	34
Figure 43: Submit and activate the irrigation rule	34
Figure 44:Overview of Synfield API in swagger	35
Figure 45: Authentication through user credentials	36
Figure 46: Description of the endpoint that fetches the list of fields of the user	36
Figure 47: List of fields	37
Figure 48: Description of the web service that fetches the list of sensing services of the node	38
Figure 49: List of sensing services per node	38
Figure 50: Description of the web service that fetches the list of measurements per sensing service .	39
Figure 51: List of measurements per sensing service	40
Figure 52: Description of the web service that fetches the list of measurements per sensing servi date range	ce in 40
Figure 53: List of measurements per sensing service in date range	41
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range	41 ge 42
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range	41 ge 42 43
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node	41 ge 42 43 e per 44
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node	41 ge 42 43 e per 44 45
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node	41 ge 42 43 e per 44 45 46
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node	41 ge 42 43 e per 44 45 46 47
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode	41 ge 42 43 e per 44 45 46 47 48
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode	41 ge 42 43 e per 44 45 46 47 48 48
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode Figure 62: Description of the web service that sets the actuator in manual mode	41 ge 42 43 e per 44 45 46 47 48 48 49
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode Figure 62: Description of the web service that sets the actuator in manual mode Figure 63: Set the actuator in manual mode	41 ge 42 43 e per 44 45 46 47 48 48 49 49
Figure 53: List of measurements per sensing service in date range	41 ge 42 43 e per 44 45 46 47 48 48 48 49 49 50
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode Figure 62: Description of the web service that sets the actuator in manual mode Figure 63: Set the actuator in manual mode Figure 64: Description of the web service that activates the actuator (manual mode) Figure 65: Activates an actuator	41 ge 42 43 e per 44 45 46 47 48 48 49 49 49 50 50
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode Figure 62: Description of the web service that sets the actuator in manual mode Figure 63: Set the actuator in manual mode Figure 64: Description of the web service that activates the actuator (manual mode) Figure 65: Activates an actuator Figure 66: Description of the web service that allows the registration of metrics by device type device serial number	41 ge 42 43 e per 44 45 46 47 48 48 49 49 50 50 e and 51
Figure 53: List of measurements per sensing service in date range	41 ge 42 43 e per 44 45 46 47 48 48 49 49 49 50 50 e and 51
Figure 53: List of measurements per sensing service in date range Figure 54: Description of the web service that fetches the list of measurements per node in date range Figure 55: List of measurements per node in date range Figure 56: Description of the web service that fetches latest measurement of each sensing service node Figure 57: List of latest measurements (per sensing service) per node Figure 58: Description of the web service that fetches the list of actuators in a node Figure 59: List of actuators per node Figure 60: Description of the web service that sets the actuator in automatic mode Figure 61: Set the actuator in automatic mode Figure 62: Description of the web service that sets the actuator in manual mode Figure 63: Set the actuator in manual mode Figure 64: Description of the web service that activates the actuator (manual mode) Figure 65: Activates an actuator Figure 67: Register measurements for a specific Plant-O-meter device Figure 68: Retrieve the measurements from a Plant-O-Meter device	41 ge 42 43 e per 45 45 46 47 48 49 49 49 50 e and 51 51 53

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



Figure 70: Virtual Fort Knox Architecture	54
Figure 71: VFK cell concept with locally hosted cells and publicly hosted cells in the main infrastru	ucture
Figure 72: Communication Process of the Manufacturing Service Bus	56
Figure 73: Communication Pattern of the Manufacturing Services Bus	56
Figure 74: Self-description of services	57
Figure 75: Exemplary pattern for data transfer of smart object to an application	57
Figure 76: Extended self-description for RESTful API	59
Figure 77: REST endpoint to register application	60
Figure 78: REST endpoint to register Smart Object	60
Figure 79: Manual app creation wizard – Step 1: Basic Information	62
Figure 80: Manual app creation wizard – Step 2: Endpoints	63
Figure 81: Manual app creation wizard – Step 3: Functions	63
Figure 82: Manual app creation wizard – Step 4: Response Events	64
Figure 83: REST endpoint to send data	65
Figure 84: Activate new Component in GUI of MSB	65
Figure 85: Security Token and Visibility Setting (MSB GUI)	66
Figure 86: Detailed Information about Component (MSB GUI)	67
Figure 87: Detailed View on Component - Configurations Tab	67
Figure 88: Detailed View on Component - List of all associated Flows (MSB GUI)	68
Figure 89: Simple Information Flow modelled in the MSB GUI (only part of GUI is shown)	69
Figure 90: Steps to compete the Exchange of Information	69
Figure 91: First Step in Creating a new Integration Flow (MSB GUI)	70
Figure 92: Drag and Drop of Components to initiate the Creation of a new Integration Flow	70
Figure 93: Selection of Event or Function from Drop Down Menu	71
Figure 94: Selection of Return Event based on preselected Function	72
Figure 95: Successfully linked Components	72
Figure 96: Detail View for Mapping Event Data to Input Parameters for a Function	73
Figure 97: View to set Condition which incoming Events are forwarded to the next Component Flow.	in the 74
Figure 98: Branching Integration Flow where one Event is forwarded to Two Components	75
Figure 99: Merging of Branches in an Integration Flow	75
Figure 100: Integration Flow after Saving	76
Figure 101: Integration steps	77
Figure 102: Synfield Platform as Application in MSB	78
Figure 103: POST request to register Synfield Platform as application to MSB	79
Figure 104: Integration flow to request data from Synfield Platform	79
Figure 105: Nitrogen trial setup	81



Figure 106: Sensors topology	.82
------------------------------	-----



List of Abbreviations

- API Application programming interface
- DIH Digital Innovation Hubs
- HN Head node
- PN Peripheral node
- JSON JavaScript Object Notation
- GPS Global Positioning System
- NDVI Normalized Difference Vegetation Index
- GUI Graphical User Interface
- VFK Virtual Fort Knox
- IPA Institute for Manufacturing Engineering and Automation
- SDK Software Development Kit
- MSB Manufacturing Service Bus



1 Introduction

The DIATOMIC project vision is to establish a sustainable ecosystem, which will facilitate digital innovation in the health, agrifood and manufacturing sectors, all of which are under-digitized and of prime importance for society and the economy. It should be underlined that such achievement is quite important for the European industry, especially the SMEs, as most SMEs and midcaps in these sectors are characterised by a low Digital Innovation Index (we name these enterprises "non-tech" hereafter).

At the heart of the DIATOMIC ecosystem, three interconnected sector-specific Digital Innovation Hubs (DIH) pulsate to accelerate digitization in-beat with sector-specific needs, and to enable delivery of AME/SSI based applications to a critical mass of customers. DIHs first assist non-tech companies to find and couple with counterparts from across the EU with complementary technological competencies. Then, acting as a one-stop-shop, they offer the teams both (i) strong technological support to accelerate design, development, prototyping and manufacturing; and (ii) business support to develop solutions with a robust product/market fit, and chart the path to market growth and investment.

This document presents a thoughtful guide on how to integrate an existing solution with DIATOMIC farm monitoring platform. The platform consists of the BIOS Plant-o-meter system and the Synfield platform and contains several components. The users and the developers may integrate their solutions with all or a subset of the described components, which best suit their needs.

For the purpose of this document, a full integration with all the components is described, so users can have a broader view of all the potentialities of every component and the greater value added by integrating then together.

1.1 Goal

The objective of the Agro Application Experiments in DIATOMIC is to create a fully functional service that offers the opportunity to farmers/agronomists/farm logistics managers to access all important parameters to accomplish efficient and self-sustainable agricultural production and offer innovative and specialized services in the agricultural domain. DIATOMIC integrates and extends in this experiment Synelixis SynField, BIOS Plant-O-Meter and Fraunhofer VFK systems to offer unprecedented flexibility to prospective developers and an enhanced experience to technology adopters and users.

SynField offers solutions that help farmers control their farming procedures remotely, with high accuracy and at low cost to meet the needs, not only of large, but also, of small and medium sized farms. SynField allows users to monitor environmental conditions at one or multiple points on the farm (installing SynField Peripheral Nodes-SF PN and one SynField Head Node - SF HN) and use them to control (through relays or actuators) **irrigation**. The sensed data are delivered to the SynCloud (through SF HN) where commands for actuations are decided. The system (currently TRL-9) also alerts the farmer under specific user-defined conditions so that they may take appropriate action.

BIOS has developed a solution for **effective nutrient balance management** in precision agriculture and gardening, through the integration of an innovative instrument for on-the-fly nitrogen monitoring into a versatile and low-cost system for expert support to farmers through a combination of disciplines like soil science, agronomy, economics and information technology. The *Plant-O-Meter is a* handheld device (currently TRL 7): the optical sensor head contains a multispectral light source, photodetector and basic electronic circuitry for signal amplification and power supply. The sensor is connected with a smartphone using Bluetooth or Wi-Fi and thus supports the farmer/agronomist to define the needs for fertilizers. The sensor also contains a high-capacity, rechargeable, Li-Ion battery to support the smartphone via USB, thus ensuring 8-hours operation of the phone.





Figure 1: Agro Application Experiment

1.2 Test scenario

Taking into account that in any precision agriculture system, prospective developers are interested to experiment with different sensors and apply different logics and algorithms, in the framework of DIATOMIC, the following functionality could be considered as test scenarios:

- Integration of Plant-O-meter with the SynField system to offer an opportunity to farmers/agronomists/farm logistics managers to monitor both environmental and soil conditions from a single point of access. The readings from Plant-O-Meter are transferred and stored at the SynField Cloud through an interface that will be developed. By accessing the SynField platform through any device (laptop, smartphone, tablet), they will be able to monitor all conditions and control irrigation and fertilisation.
- Enrich SynField with drones equipped with cameras that fly over the fields and collect visual information enabling the farmers to inspect every square meter of their farm without physically visiting it. The collected information may be delivered to the farmers through the SynField front end.
- Integrate an advanced version of the Plant-O-Meter instrument onto a mobile robotic platform (to allow for automation of the data collection process and identify two pilot sites most suitable for testing and validation of the automated version of the solution.
- Integrate SynField Cloud with Virtual FortKnox towards delivering to developers a platform that accelerates significantly application development. While SynField enables the user to set rules for notifications and actuation (e.g. alert me when soil humidity is below a threshold), more advanced applications like disease prediction necessitate more advanced algorithms execution on the collected data and possible combination with 3rd party sources, using the VFK platform.

In the Annex of this deliverable some trials are described in detail.



2 Modules description

In this section, the different modules are presented and described. This includes the overall description of the functionalities that each module contains with all the necessary information that a user may need to understand and start working with them.

2.1 The Plant-O-Meter smartphone application

BioSense has developed a portable multispectral optical device called Plant-O-Meter (currently at level TRL 7) for precise plant stress measurement, which reconciles the requirements for a simple, costeffective optical device and accuracy of applied optical method. The proposed device is based on a specially designed multispectral source that integrates the light sources of the four most indicative wavelengths in one optical module and enables simultaneous illumination of the whole plant. The four wavelengths emitted by the source, namely 850, 630, 535 and 465 nm, are chosen to cover the most indicative spectral bands and enable the calculation of multiple vegetation indices. Illumination and detection of reflected signals are performed in a sequential manner, which provides very quick measurements and raw data collection. The detector module sequentially receives the reflected spectrum and sends a raw data to the microcontroller. Each measurement comprises 128 excitation impulses for each wavelength, and the reflected signal is calculated as an average of the corresponding reflected pulse train. The raw data is transmitted to and processed in a smartphone using a specially developed application, which enables determination of vegetation indices. In particular, signifying plant stress caused by nitrogen deficiency and/or drought. The operating principle of the Plant-O-Meter is shown in Figure 2.



Figure 2: Operating principle of the Plant-O-Meter

In the frame of the DIATOMIC project, a new version of the device has been developed that is more immune to the ambient lighting and more ergonomic than the previous one. The updated version together with the smartphone application is shown in Figure 3.



Figure 3: Plant-O-Meter (left) and specifically designed smartphone application (right)



One of the key advantages of the proposed solution for plant stress and nutrient measurements is its coupling with the smartphone application. With the help of the application, one can initiate, gather, georeference and save data. Furthermore, the application uploads measurement results together with GPS coordinates to the SynField API (see section 2.2.11). The main menu of the application is shown in Figure 4. Below the figure, each menu item is explained separately.



Figure 4: Main menu of the Plant-O-Meter application

The first menu item *Find devices*, initiates Bluetooth connection between the Plant-O-Meter and the smartphone. Once the connection is established the user can choose between two single point modes and one automatic mode of measurements. The first mode, Figure 5, is made for point measurements at an arbitrary location. In this mode the user can initiate measurements, save them, and upload them to the SynField API if there is an internet connection. For safety reasons all measurements data are stored in the smartphone's memory until the upload to server is successful. Arbitrary location mode offers instant optical indices results.



Figure 5: Point mode of measurement at an arbitrary location

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



The second point mode of measurement is related to the Head Node of SynField cloud platform. The user should be at the location of the node and initiate the "*Generate Points*" procedure which will create measurement points distributed over homocentric circles with a radius of 5, 10, 25 and 50 meters from the node. On each radius, a total of 8 point measurements will be done with the angle shift of 45°. It is also left as an option for the user to choose the number of circles, radius of circles and number of points per circles. These canopy measurements will estimate the health status of the plant surrounding the nodes with respect to nutrients levels based on vegetation indices provided by the device. All the point measurements are automatically related to the appropriate node and uploaded to the SynField cloud. In this mode the user is able to save the data of measurements.



Figure 6: Head node homocentric circles mode of measurement

The last measurement mode is called *Automatic measuring* and is dedicated but not restricted to automatic methods of measurements related to the mobile robotic platform Clearpath robotics Husky A200, tractors or similar agriculture machinery. In this mode the user is able to take continuous measurements which means having the ability to start logging by pressing START and automatically store the georeferenced measurements (points) when pressing STOP. Automatic measurements are done every second and stored in the smartphone's memory until the STOP button is pressed. It is possible to continue with the measurements by pressing START again. An option to initiate a new set of measurements results is created in real time while the logging is done. Each point measurement is visualized with different colours according to the Normalized Difference Vegetation Index value.

The menu item named Send data is sending the results of measurements to the SynField API. After the successful sending, the data will be permanently deleted from the smartphone's memory.

The data exchange between Android application and SynField cloud platform is achieved thought the Synfield API. The data format is the JSON one while its structure is shown below:

{
 {
 "red":"string",
 "green":"string",
 "blue":"string",
 "infrared":"string",
 "latitude":"string",
 "longitude":"string",
 "date":"dd.mm.yyyy. hh:mm:ss"
 }
]

More details about the integration are depicted in section 2.2.11.11.





Figure 7: Continuous mode of measurements

The Android application can send on SynField API one or multiple measurements where each measurement is described by the above-mentioned attributes. Reflectance values of red, green, blue, infrared are presented by attributes "red", "green", "blue" and "infrared" respectively, while GPS coordinates are stated by "latitude" and "longitude" attributes. Finally the timestamp of measurement containing the date and time in the format <day.month.year hour:minute:second> is stated by "date" attribute.

The Settings option offers the user to choose a different language. For now, English and Serbian are available. The last option of the main menu is Exit which closes the application.



2.2 Synfield Platform

The Synfield platform is a cloud-based platform implemented by Synelixis. Given the Synfield platform, any user is able to register a field and setup the network of the installed Synfield nodes (devices) on it. Synelixis is the manufacturer of two types of nodes: the head node that is the gateway towards the Internet and the peripheral node that is able to communicate only with the head node that belongs in the same network. Figure 8 presents an indicative network of nodes (in the left side) and how these are connected. Typically, each field contains one head node and optionally, one or more peripheral nodes. The head-to-peripheral (and vice-versa) communication is performed using RF technologies while the head node is able to communicate (upload measurements, retrieve configuration and commands) with the Synfield platform (REST API) through WIFI or GPRS/GPS.



Figure 8: SynField ecosystem

Each node, head or peripheral one, can retrieve measures from one or more wired installed sensors. The exact number of the possible installed sensors on each node depends on its type, head versus peripheral, and its hardware version. Synfield nodes are compatible with several types of sensors; the only limitation is the interface of the sensors. Currently, ADC, PULSE, I2C and SERIAL sensors are fully supported using RJ connectors. Apart from the sensors, the Synfield nodes are compatible with a set of actuators. Also, in the frame of DIATOMIC project, the Synfield platform has been extended so as to support the manipulation of external devices such as the Plant-O-Meter device (provided by Biosense). As external devices we consider the devices that are not compatible with or not plugged in the Synfield nodes.



Figure 9: SynField Head Node (left) and Peripheral Node (right)

As shown in Figure 9, SynField nodes are compact self-contained. They feature a Photovoltaic panel in order to be autonomous and a Regulated Lead-Acid (Lithium) battery to operate for days even without being charged. Moreover, the electronic circuit is located in a waterproof box (IP-67), which protects the system from rain, snow and hail.

Figure 10 provides a typical SynField installation, where one headnote and multiple peripherals in ad-hoc networking architecture are installed.





Figure 10: Typical SynField installation

2.2.1 Create a new account on the Synfield Account

The first step that a user must do is to create an account. This is offered by the DIATOMIC SynField platform instance free of charge for all users (both the beneficiaries accepted from the DIATOMIC Open interested Calls and for any party). То create the account, follow the link http://83.235.169.221:9990/en/register/ and type your personal information, check the box "I'm not a robot", accept the terms of service and proceed on the account registration.

SynField					🗱 English 🗸	Log in	Sign up
		Create All fields a	Baccount re required				
	synfield	o ^e .	mall@domain.gr	@			
		24		2 _a			
	Peccetation		PASSWORD MAT	жн			
	Synfield	L,	User	٩,			
	Greece	Ŷ	Europe/Athens	÷			
	I'm not a robot	TeCAPTCH/ Promy-Terms					
	Subscribe to Synfield n Accept terms of service	ewsletter					
		•	Back to login form	e account +			

Figure 11: Registration in the Synfield platform



2.2.2 Account upgrade

When a user accesses for the first time the Synfield platform, they are assigned a free plan (no fee). That means that the user can not register any Synfield device in the platform, but just receives information from selected already available nodes.

Whenever needed, the beneficiaries of the DIATOMIC project can upgrade their SynField account (without fee) to a plan that allows the registration of the Synfield devices (including sensors and actuators) in the platform, use the API and enable a set of services that SynField platform provides. Contact us at <u>synfield@synelixis.com</u> in order to upgrade your account for free. After the completion of the DIATOMIC project, any upgrade will expire.



Figure 12: History of user subscriptions



Figure 13: Available SMS and remaining API calls



2.2.3 Register a field

For the collection and the storage of any kind of measurements provided from Synfield (head and peripheral) nodes and external devices (e.g. Plant-O-Meter), their registration in the Synfield platform is necessary. Within DIATOMIC, the SynField nodes will be already installed, thus no further action will be needed. For completion, we also provide the relevant information.

The registration of a new field takes place in the Configuration>Fields page. There are two options to register a field including one or more Synfield nodes in the SynField platform. The first option is to configure the Synfield devices (that will be located in the area of interest) through the SynControl android application (<u>https://play.google.com/store/apps/details?id=com.synelixis.SynControl</u>) and upload the configuration file in the platform through the button *Import configuration*.

					+++ 🕸 🗢 🗢 DDD vodal	Center 1978 (5 G F # 1977)		\$ 🗢 ====0	vodafene GR 3G 4
SynControl firmware connect		INECTED	<i>←</i>	Configure			SynControl firmware control	v con	NECTED
4	₿	2		SENSORS	ADVA	NCED		8	2
CONTROL	MONITOR	REMOTES	ANAL	OG SENSORS			CONTROL	MONITOR	REMOTES
			DIGIT	AL SENSORS			Remote 1 (IC	RF)	2
-	CONFIGURE		Digita	I channel 0 se	nsirion_sht1x	_	Remot	e #1	
			Digita		ne		÷	BLINK ONCE	
1 E	LINK SYNFIEL	LD	AOTI	ATODO			Non-		
		NNECT	Actua	tor 0 Jatching	valve		•	START BLINKI	16
C PIRIO	IWARE DISCO	NNEGI	Actua	tor 1 latching	_valve		8	REMOVE REMO	TE
-0 R			Actua	tor 2 none					
			Actua	tor 3 none					CLOSE
Э в		ION							
									(+)
		(
		LOG				SEND			LOG

Figure 14: Selected SynControl Android app screen

The second option is to register by clicking on the button *New Field* as shown in Figure 15. The user needs to set the name of the field, its coordinates and the type of the crop as depicted in Figure 16.







SynField	=	🔛 English 🗸 🐺 📿 🎍 synfield 🗸
Synfield User O	We Home / Configuration / Fields / Add new field	
MADE	Add new field	
Measurements >	Set the coordinates of the field	Field name *
Reports >	Satelite Cossa Cros	MyField
😹 Diseases >		Longitude *
11 Management >	Katerasa Laissa	22.3653
🏟 Configuration 🗸	Kozukas Prices Adoleco	Latitude *
	Elan Palamas	39.5168
Overview	Mouzaka Moučaki Vistore	Crop type *
Fields	Karobiren Sofacen Especies Faralo +	Lemons *
Nodes Network >	Google	
Synfield Sensors	Map trata \$2018.Google Imagery \$2018.TiertaMetrics Terms of Dae	
Synfield Actuators	(*) Required netos	Cancel Submit
External Devices		6
🖬 Calendar		
(=) News		
🖶 Shop		
Help Center		

Figure 16: Register in the platform a new field

After the successful registration of the field, the field management as well as the registration of Synfield head nodes can be done as shown in Figure 17.







2.2.4 Define your network of Synfield devices

Navigate in the *Configuration>Nodes Network>Head nodes* to register a new head node assigning in a field or manage the existing ones. Click on the *New node* button to process on the registration of the head node (see Figure 18).

SynField	=	🖼 English -> 🐺 😡 🚨 synfield ->
Synfield User	R Home / Configuration / Hoad nodes	
MAIN	Filters 🕐	•
🗽 Measurements 🔹 🔸	MyField × ~ X Clear all	
🔓 Reports >		
🕷 Diseases 🔹 🔸		
Management >	My SynField head nodes	+ New node
🔅 Configuration 🗸		
Overview		Block EList
Fields	 No head nodes were found. 	
Nodes Network 🗸 🗸		
Head nodes		
Peripheral nodes		
Synfield Sensors		
Synfield Actuators		
External Devices		

Figure 18: Register a Synfield head node

To complete the registration, fill in the required details in the wizard. In the tab Description, select the field of interest, type the serial number of the node (provided from Synelixis), select the hardware version of the node and type other details as shown in Figure 19.

SynField	=			GIS English ~	W Q	🛔 synfield 🗸
Synfield User	Home / Configuration / Head nod	es / Add head node				
	Add head node					+ Cancel
🖌 Measurements >						
Reports >		٢				
🕷 Diseases 🔹 🔸	1	Provided by		Coordinates		
1 Management >	Field *	Synelixis	GSM	t via "		
🔆 Configuration 🗸	Node name *	/	SIMICC			
Oveniew	MyField	0	required for	GSM		
Fields	Serial number * ()	7	Phone number			
Nodes Network 🗸 🗸	ef 12:34:12:ad 33		required for (GSM		
Head nodes	Hardware version *					
Peripheral nodes	Synfield v6.0					
Syntield Sensors	(*) Required fields					
Synfield Actuators					Previous	Next
External Devices						

Figure 19: Step 1- Type the basic details of the Synfield head node



In the tab Coordinates, define the exact coordinates in which the head node has been installed and press submit (Figure 20). Repeat the aforementioned steps to register more head nodes.



Figure 20: Step 2 - Modify the head node's coordinates

In analogous manner, the user can register a Synfield peripheral node as part of the network of the Synfield nodes. The only difference here is the fact that the user needs to associate the peripheral node with one of the existing head nodes (see Figure 21). In this way, the user defines a network of Synfield devices that covers the area of interest. The head node is the representer of the network since it is the one that has access to the internet.

SynField	=	IIII) English ~ 🦞	Q 🛔 synfield -
Synfield User Ø	Home / Configuration / Peripheral nodes / Add peripheral n i	ode	
MARK.	Add peripheral node		- Cancol
🖌 Measurements >			
Reports >			
∭e Diseases →	Description	Coordinates	
∭ Management →	MyField	• ef 12:34 12 ed 33_1	
🔅 Configuration 🔶	Node name "	Hardware version *	
Overview	Peripheral node #1	SymRemote_v1	
Fields	(*) Required fields		
Nodes Network 🛛 🛩		Serial number of	Province Next
Head nodes		peripheral node	
Peripheral nodes			
Synfield Sensors			
Synfield Actuators			
External Devices			

Figure 21: Register a peripheral node as part a Synfield network





Figure 22: Manage the nodes in similar way

After the registration of the head and peripheral nodes in the platform, the user is able to manage (view, edit, delete) them as depicted in Figure 22.

2.2.5 Register sensors in existing Synfield nodes

The next step includes the registration of the plugged-in sensors in each Synfield node. To do this, navigate on the *Configuration>Synfield sensors* area and press the button *New sensor* (Figure 23). By default, a set of sensors, that reflects the health of the node (battery and so on), is registered automatically.

SynField	=				33 Er	nglish - 🗑 🖇	🔉 🎍 synfield
Synfield User	N Home / Configuration	n / Sensors					
MAIN	Filters ()						~
₩ Measurements >	Select a field fro	m the list 🗸	Select a node from the list	- Select a sense	or type from the list 🛛 👻	× Clear	
Reports >							
💥 Diseases 🔹 🔸							
II Management	My sensors						+ New sensor
🔅 Configuration 👻							
Overview	Search:	Q				Columns ~ S	ihow 10 ~ entries
Fields	Field ^	Node ^	Sensor 0	Manufacturer 0	Service(s) 0	Connector ^	Actions 0
Nodes Network >	MyField	MyField	Solar panel current	General	Current		≡·
Synfield Sensors	MyField	MyField	Temperature (internal LM61)	General	System temperature	ж	≡-
Synfield Actuators	MyField	MyField	Solar panel voltage	General	Solar radiation level	2	≡.
External Devices	MyField	MyField	Battery voltage	General	Battery	1	≡-
🛗 Calendar	MyField	Peripheral node #1	Battery voltage	General	Battery		= -
(+) News	MyField	Peripheral node #1	Solar panel current	General	Current		≡-
🚍 Shop	Showing 1 to 6 of 6	entries				P	revious 1 Next
Help Center							

Figure 23: Add a new sensor in the head node

The registration of a sensor includes the selection of its type, the associated (head or peripheral) node, the connector in which the sensor is plugged and optionally the adjustment of some metadata (e.g. soil type and sensor depth in case of soil moisture sensor) as depicted in Figure 24. Figure 25 depicts the registered sensors with further details such as the sensing services that each sensor measures; the management actions of the sensors include view, edit, rename sensing service and delete while the filtering of sensors per field, node or type is available.

For the agrifood experiment, the collection of sensing services such as air temperature, relative humidity, soil moisture, wind speed, wind direction, rainfall, flow meter will be meaningful.

2.2.6 Register actuators in existing Synfield nodes

The next step includes the registration of the plugged-in actuators in each Synfield node. To do this, navigate on the *Configuration>Synfield Actuators* area and press the button *New actuator* (Figure 26). The registration of an actuator includes the selection of its type, the associated (head or peripheral) node, the connector in which the actuator is plugged on (see Figure 27). Figure 28 depicts the registered actuators with further; the management actions of the actuators include view, edit, rename service and delete while the filtering of actuators per field, node or type is available.

A kind of actuator that will provide added value in the experiment is the solenoid valve.





SynField			III English ·	> ₩ Q synfield ~	
Synfield User	ration / Sensors / Add new sensor				
Add new se	ensor			← Cancel	
<u>لا</u> Measurements					
Reports		Sensor type			
₩ Diseases →	Sensor type	Node	Connector	Properties	
Al Management		N		One and a sector	
Search sn	P.	1		Snow 10 v entries	
	Sensor type	 Monufactu 	er C Service(s)	0	
Overview	Relative Humidity/Temp. (SHT1x)	Sensiri	on Air temperature, Ai	ir humidity	
Fields	Relative Humidity/Temp. (SHT2x)	Sensin	on Air temperature, Ai	ir humidity	
Nodes Network	Relative Humidity/Temp. (SHT3x)	Sensiri	Air temperature, Ai	ir humidity	
Synfield Sensors	Relative Humidity/Temp. (SHTCW)	() Sensiri	Air temperature, Ai	ir humidity	
Syntield Actuators Showing 1 to	4 of 4 entries (filtered from 35 total entries) 1	row selected		Previous 1 Next	
Exemai Devices					
🛗 Calendar				Previous	🛔 syntield ~
Syntield User	Home / Contiguration / Sensors / Addine	wsensor			
	Add new sensor				← Cancel
MAIN					
₩ Measurements >		0	Define node	0	
Reports >	Sensor Ivna	Node	Connector	+ Properties	
X Diseases	Serior type	Nuce	Connector	Fruperses	
8 Management	Search: Q			Show	10 - entries
🏘 Contiguration 👻	Field	No		CAL A	
Overview		M		975 V	
Fields	MyField	wyneso		ef. 12:34:12;ad:33	
Nodes Network	MyField	Peripheral node #	PERIPHERAL NODE	ef 12:34:12 ad 33_1	
Synfield Sensors	Showing 1 to 2 of 2 entries 1 row select	ted		Previous	1 Next
Synfield Actuators					
Contract Contract					and the second se

Figure 24: Follow the wizard to complete the sensor's registration

SynField	=			El Eng	lish 🗸 🦉 😡	🕽 🔺 syntieid 🗸
Syntield User	Home / Configuration / Sensors					
AMAIN .	Filters 👩					v
₩ Measurements >	Select a field from the list	Select a node from the list	 Select a sen 	isor type from the list 🗢	¥ Clear	
Reports >						
)∰, Diseases →						
t∭ Management >	My sensors	The new sensor has bee	en			+ New seasor
Configuration •	1	olugged in the 8th conne	ctor			
Overview	Search				Columns v Sh	iow 10 - entries
Fields	Field A Node A	Sensor	Manufacturer C	Service(s) C	Connector *	Actions 0
Nodes Network >	MyField MyField	Solar panel current	General	Current	2	
Synfield Sensors	MyField MyField	Temperature (Internal LM51)	General	System temperature	(1	
Synfield Actuators	MyField MyField	Battery voltage	General	Battery	12	
External Devices	MyField MyField	Solar panel voltage	General	Solar radiation level	÷	≣-
🛗 Calendar	MyField MyField	Relative Humidity/Temp. (SHT1x)	Sensirion	Air temperature, Air humidity	8	≡·
In News	NyField Peripheral node #1	Battery voltage	General	Battery	*	≡-
🖀 Shop	MyField Perpheral node #1	Solar panel current	General	Current	2	
11 Help Center	Showing 1 to 7 of 7 entries				Ph	evinus 1 Neci

Figure 25: Manage the registered sensors



SynField		=								5	🛿 English 👻	₩	Q 🛔	synfield -
Synfield User	0	# Home / Config	juration / Actu	tors										
MAIN		Filters ()												~
₩ Measurements		Select a fil	eld from the list	÷	Select a n	ode from th	e list 👻	Select a	type from the list	Ū.	× Clear			
Reports	3 * -:													
🗮 Diseases	.*												_	
0] Management	•	My actuate	ors								+ News	schuator	Manageme	nt panel
Configuration														
Overview		Search		Q.							Column	is v	Show 10 ~	entries
Fields		Field	• No	10 ^	Actuator	¢	Manufacturer	\$	Service(s)	¢	Connector	•	Actions	c
Nodes Network							No data avail	lable in table	i.					
Synfield Sensors		Showing 0 to () of 0 entries										Previous	- Nave
Synfield Actuators													111111	
External Devices														



SynField	=			🖽 English - 🦞 😡	🛔 synfield 🗸
Synfield User	# Home / Configuration / Actuators /	Add new actuator			
	Address				
MAIN	Add new actuator				e- cancer
₩ Measurements >	~				
Reports >	٢		(2)	(1)	
💐 Diseases 🔅 🤌	Actuator typ	Step 1: Select	t type	Connector	
👔 Management >	Search	of actuate	or	Show 10 -	- entries
Configuration 👻			< <u><</u>	120000000000000000000000000000000000000	
Overview		Sensor type	Manufacturer	C Service(s) C	
Fields		General relay	Genetic	Motor	
Nodes Network		Solenoid Valve	Genenc	Irrigation	_
Synfield Sensors		Solenoid valve (Latching)	Generic	Irrigation	
Synfield Actuators		Solenoid valve 2 inch (Latching)	ReinSpA	Irrigation	La synfield ∽
External Devices	Showing 1 to 4 of 4 entries 1 ro	w selected		Previous	Nut
🗰 Calendar					Cancel
				Previous	Next
🔛 Measurement	s ()	0			
Reports	×.	Actualitor type	Node		Connector
😹 Diseases	<u>ر د</u>		5	ten 2: Define node	
1 Management	Search:	Q	5		Show 10 - entries
🗱 Configuration	· + .	Field C	Node	C SN	0
Overview		NuField	MyField Revolution	of 12 34 12 a	d 33
Fields			Dariphasel and att		
Nodes Netwo	* *	MyField	- autoretai none au	ef 12:34:12:ad	133_1
Synfield Sens	ors Showing-1	to 2 of 2 entries 1 row selected			Frevious 1 Next
Synfield Actua	lors				
External Devi	pes :				Previous Next
External Devi	tes .				Previous





SynFiel	ld		Ξ										ILIER E	English + '	₩	Q 🛔 🕯	yntield 🐱
	Synfield User	٥	W Home / C	onfiguration	I / Actuato	HG.											
MAIN			Filters	Ð								Manage ma	the ac anually	tuator			~
Lee Me	sasurements	2	Select	a field from	the list	~	Select a node from	the list	~	Selec	t a type fro	m the list	~	× Clear			
Re	ports													-	_		
🕷 Di	seases	•	-												-	>	
ei Ma	anagement	- iii	My acti	uators			The n	ew act	tuator	has b	een			+ New ach	sator	Managemen	t panel
🔅 Co	miquration	*					plugg	ed in th	he 6" c	conne	ector						
OV	renview		Search.			Q			1					Columns	~	Show 10 ~	entries
Fie	ekts		Field		Node	•	Actuator	0	Manuta	cturer	0	Service(s)	0	Connector	•	Actions	\$
No	ides Network		MyF	Teld	MyFiek	đ	Solenoid Valve (Latchin	g)	Ge	meric		Irrigation		6		,≣·	
Sy	nfield Sensors				and a												
Sy	nfield Actuators		Showing 1	to 1 of 1 er	ntries											Previous	Ned
Ex	ternal Devices																

Figure 28: View the registered actuators

2.2.7 Register external devices

Synfield platform supports the management of external devices by clicking on the tab *External Devices* in the left menu. A new device can be registered in the platform by clicking on the *Add device* button.

SynField		=										GIB EN	glish ~	W	Q	å 5	intield ~
Synfield User	۵	# Home / Conte	guration	f Edemal de	vices												
ман		Filters ()															. •
Measurements	\$	Select a fi	eld from t	he list		Select a device type	from the list	÷	× Cle	łr							
Reports	×																
🗯 Diseases	ė														-		
II Management	5	Installed e	externa	l devices												F Add	device
Configuration	-																
Overview		Search			Q								Column	s ~	Show	10 ~	entries
Fields		Field	•	Device	•	Serial Number	•	Manufa	acturer	0	Service(s)	\$	Date	0	A	ctions	0
Nodes: Network	×						N	o data avail	able in tabl	e							
Synfield Sensors		Showing () to	0 of 0 entr													Taurour	Next
Synfield Actuators		anowing 0 to	n or A eut														hierd.
External Devices																	

Figure 29: Add a new external device (e.g. Plant-O-Meter)

A device can be registered easily and rapidly by selecting the field in which the device is located or planning to be installed, the type of the device and by typing the serial or product number as shown in Figure 30.



SynField	=	ඎEnglish ~ 몇 Q 🎍 synfield ~
Synfield User	We Home / Configuration / External devices / Register	
MAIN	Register new device Select the type of supported devices	
₩ Measurements >	Select the proper field *	
Reports >	MyField	
💥 Diseases 🛛 🔸	Select the type of device *	
0] Management >	Plant-O-Meter	
🔅 Configuration 🗸	Serial Number *	
Overview	(*) Depuided fields	
Fields	Cancel Submit	
Nodes Network >	-	
Synfield Sensors	lype the serial number of the device	
Synfield Actuators		
External Devices		

Figure 30: Register the Plant-O-Meter in Synfield platform

After the successful registration of the device, more details per device are depicted in the management page (see Figure 31). The preview, edit and deletion actions of each device are supported through the Actions drop-down list.

SynField	≡ 8	III English	1 ~	W	9	🛔 5y	nfield v
Synfield User	W Home / Configuration / External devices						
MARIN	Filters ()						*
₩ Measurements >	Select a Baid from the list - Select a device type from the list - x Clear						
Reports >							
🗶 Diseases >							
8 Management	Installed external devices					F Add	device
🌣 Configuration 🐱							
Overview	Search: Q.	c	Columns	4	Show	10 ~	entries
Fields	Field Device Service(s) Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field Field	0	Dat	le (0	Actions	0
Nodes Network >	MyField Plant-O-Meter KON112212(2) Biosense Normalized Difference Vegetation Ind	lex.	2018	-05-15	i		
Synfield Sensors					2.7		
Synfield Actuators	Showing 1 to 1 of 1 entries				Previo	5 1	Next
External Devices							

Figure 31: Manage the registered devices

2.2.8 Data visualization

SynField allows for viewing of the current measurements of all the sensors attached on the user's nodes from the *Measurements> Current status* on the left menu. The node can be chosen on the left of the page and the last recorded values will appear on the right in separate frames per origin: sensors' measurements, calculated metrics, actuators, internal system measurements. Normal values appear green while red colour denotes an issue.



Figure 32: Current status of sensing services in Synfield node

Each panel carries specific information: the name of the service, the value, the date and timestamp of the recorded value, the connector the sensor is attached to the node. Each service is depicted by a certain icon found in the left top corner of the panel and on the little flag icon each service may be pinned on the user's dashboard.

SynField also carries a powerful visualization tool in the *Measurements>Plots* page. Clicking on the *Add new* button, the user is able to compose a plot of interest by combining sensing services that are measured from devices installed in the user's fields.

				- Add Nevi Save 🗃 🔤		
			Select sensor data to plot	include forecast 🌍		
	(*) (*)	🔛 Γραφημα 1	v 🗌 Monemvasia Mine		1 Choose a date range	⊠ ° • ×
Connect status			> Monemvasia PN 2			
			Monemvasia PN 1	Visualize forecast		10
Piots			✓ Monemvasia			12
Length Contraction		27	Battery (Connector 0)			*
	<u> </u>		System temperature (Connector 0)		V 7	
	<u>8</u>		Solar radiation level (Connector 0)			
			Current (Connector 0)			
			Wind direction (Connector 7)			018
-Configuration			Rain (Connector 7)			
			Wind speed (Connector 7)			
			Soil moisture (Connector 2)		4	E.4
			Soil moisture (Connector 3)		- 74,	
			Air temperature (Connector 8)			
🔠 Shop			Air humidity (Connector 8)		· · · · · · · · · · · · · · · · · · ·	
		054 201	Pressure (Connector 1)		05-08 05-0 2018 201	0 8
			Evapotranspiration	Generate the plot		
			Daily GDD	1		
			GGB hetenestina		*	
Api docs			↓ Expand all Collegee all ↑	Cancel Plot Lv		
		L			·	
		Copyright @ 2013 - 2018	Synelizis Solutions S.A. All rights reserved.			

Figure 33: Select sensing services to be visualized

As soon as the node of interest is expanded all the measurements referring to it appear. SynField also allows for viewing a forecast of the corresponding weather conditions by activating the *Include forecast* option on the top right corner of the pop-up window. The Plot button on the bottom right corner of the pop-up window creates the new plot.

diatomic





Figure 34: Visualize air temperature and relative humidity in one plot

2.2.9 Diseases

SynField provides disease prediction for numerous crops though the monitoring of soil characteristics, leaf moisture and local weather conditions and their correlation to the type of crop and stage of growth. In the Disease Repository there is an extensive list of diseases SynField supports. Each appears in a separate panel where apart from a brief description, by clicking on *Affected crop types*, a list of all the crops affected by the disease appears.



Figure 35: Affected crop types by specific disease

There is a filtering tool which allows search in the repository per species or per field.

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



Figure 36: Diseases Repository

The *Monitor disease* button starts the procedure of adding the prediction of the chosen disease to your field. Clicking *Select* the page where all the necessary information to be defined appears. When all is set the Monitor button creates the disease prediction.



Figure 37: Initialising disease prediction in a field

diatomic



MAIN	Jp- Monitor field Monemvasia for Corn Earworm (Helicover	pa zea).
₩ Measurements >	Fill the following information in order to start monitoring your field.	Factors defiition
🔝 Reports >		
🗯 Diseases 🗸 🗸	Provide an air temperature sensor *	These temperature in Celaius.
	Air temperature (C) - Monemvasia (Connector 8)	12.6
Repository	Set the status of the created disease *	T _{max} temperature in Celsius.
Monitored diseases	Published	× 33.3
۱) Management >	(*) Required fields Select the proper	Initial growing degree-day value.
🔅 Configuration >	sensor	٥
🗰 Calendar		Growing degree-day calculation start date.
(*) News		a 2018-05-21
🚍 Shop		() It is best to use the default provided values for more accurate disease prediction.
🗐 Help Center	+ Back to Disease Repository	Monitor. Ar
LINKS		
Api docs		

Figure 38: Setting disease prediction details

2.2.10 Smart and simple rules

SynField also allows the creation of rules, smart or custom ones, exploiting measurements of the plugged sensors as well as plugged actuators. These rules can manage an actuator, enable alerts/notifications or calculate a kind of metadata (e.g. evapotranspiration, GDD and so on).

In the *Rules repository>Smart rules* are SynField's predefined rules. Focusing on the smart irrigation, the user is able to create a smart irrigation rule by clicking on *My smart rules* button on the Irrigation panel. SynField then creates a draft version of the smart rule which you can be adapted to the user's needs by clicking *Edit*. Figure 39 up to Figure 43 depicts the completion of the irrigation rule based on user's preferences.



Figure 39: Basic step of irrigation rule



MAIN	Ir						+ Cancel
🗠 Measurements	>						
Reports			(0)			(4)	
😹 Diseases		Basic	Settings	Crop pa	rameters	Actions	Submit
Monagement	*	Define the sensors tha	at the rule will consider		Irrigation cha	aracteristics	
Rules repository	(5 1						
My smart rules		Soil moisture sensor *			Irrigation area per dripper (m2) *		
My custom rules		Soil moisture (%) - Monem	Ivasia PN 2 (Connector 2)	*.	0.25		
My actuators		Air temperature sensor *			Irrigation drippe	r flow (litres/hour) *	
	_	Air temperature (C) - Nsµš	a (Connector 6)	÷	4.0		
Configuration		Relative humidity sensor *					
📆 Calendar		Air humidity (%) - Νεμέα (0	Connector 6)	×.			
(+)) News		Wind speed sensor *					
🖀 Shop		Wind speed (m/sec) - Nsp	ιέα (Connector 1)	¥			
FI Help Center		Solar radiation sensor *					
		Estimated solar radiation 1	from panel	2			
Api docs							
		(*) Required fields					

Figure 40: Define the metrics, necessary for the irrigation rule

	Reports)			5
	Diseases		Basic	Settings	Crop para	neters	Actions		Submit
0E	Management	~	Calculate the daily grow	ring degree days (GDD)		Adjust the follow	ing indicators o	f your crop (Pe	eppers)
	Rules repository	*	Base temperature value (C) *						End of
	My custom rules		10			Stage	Crop coefficient (Kc)	Depletion factor (p)	stage (based on growing
	My actuators	_	50	(C) *					degree-days - GDD)
\$	Configuration	3	Calculation method *			Late	0.9	0.5	2400
(m)	Calendar		Baskerville-Emin (recomme	nded)	÷	Mid-season	1.05	0.3	1750
-	Shop		(*) Required fields			Crop Development	0.6	0.2	900
89	Help Center					Initial	0.6	0.2	380
. 1.04	Ani doca					(1) SynField provides	a set of default valu	ies for your crop ty	pe
									Previous Next

Figure 41: Define the soil profile

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



itors			Node ^	Ac	ctuator	0	Connector	C	
ition		м м	onemvasla PN 1	Solenoid	Valve (Latching)		3		
		□ M	onemvasla PN 2	Solenoid	Valve (Latching)		3		
		Showing 1 to 2 of 2 entries	Showing 1 to 2 of 2 entries 1 row selected					Provioun	1 Nort
					-				
		Activate actuators on fo	ollowing time ranges (if irrig	ation is	Advanced settin	ngs for actuators	8		
	necessary)			Minimum activation	time (in minutes)				
		Startin	g time	Actions	10				
		hourminutes +		+	Maximum activation	n time (in minutes)			
					20				
	1 The time period, in which the actuators are active, is calculated from the irrigation smart rule. Therefore, the rule will adjust your input.		ed from the	Maximum deactivat	tion time (in minutes)			
					minutes				
					1 Leave the above for further details.	fields empty if you a	re not <mark>su</mark> re ab	out them or con	tact with u

Figure 42: Associate one or more actuators with the irrigation rule and define the irrigation plan

MAIN	()	Irrigation exert
₩.	Measurements	
B	Reports	
*	Diseases	Basic Settings Crop parameters Actions Submit
68	Management	Do you want to activate the rule? *
	Rules repository	Yes, I do v
	My smart rules	What type of notification do you want to receive? *
	My custom rules	All
	My actuators	Previous Submit
¢	Configuration	
Ħ	Calendar	
(•)	News	
	Shop	
EE	Help Center	
LINK		
	Api docs	Copyright © 2013 - 2018 Synelixis Solutions S.A. All rights reserved.

Figure 43: Submit and activate the irrigation rule

As soon as the *Submit* button is pressed the rule will appear in the smart rules page.

2.2.11 Synfield REST API

Synfield provides a RESTful API so as any user is able to retrieve data related to his account. Also, the Synfield API facilitates the integration among the Synfield platform and other platforms. The format of the timestamp, included in each web service, is based on the ISO 8061. Synelixis provides an interactive API documentation for the Synfield API that enables the users to try out the API calls directly in the



browser. The API documentation is based on the swagger framework. The Synfield API documentation is public available in the link <u>http://83.235.169.221:9990/en/api/docs/</u>.

💮 swagger	http://83.235.169.221:9990/en/api/docs/api-docs/	api_key	Explore
		Dja	ango REST Swagger

Documentation of Synfield web services

Find below the documentation of REST web services that Synfield provides. Measurement results are limited to 1000 per page.

Contact the developer Synelixis Solutions SA

actuat	ors	Show/Hide List Operations Expand Operations Raw
GET	/en/api/v1/actuators	Returns the Actuators that the user has access to
PUT	/en/api/v1/actuators/{actuator_id}/automatic	Updates an Actuator to Automatic Mode
PUT	/en/api/v1/actuators/{actuator_id}/manual	Updates an Actuator to Manual Mode
PUT	/en/api/v1/actuators/{actuator_id}/state/{state}	Updates the Actuator state to on/off
exterr	al-devices	Show/Hide List Operations Expand Operations Raw
GET	/en/api/v1/external-devices/{device}/serial-number/{serial_number}/	measurements
		Retrieve the measurements of the device by given its serial number
POST	/en/api/v1/external-devices/{device}/serial-number/{serial_number}/	measurements
	Register one o	more measurements related to a device by given its serial number
fields		Show/Hide List Operations Expand Operations Raw
GET	/en/api/v1/fields Returns th	e Sections that are owned by the user with the matching credentials
nodes		Show/Hide List Operations Expand Operations Raw
GET	/en/api/v1/nodes/{sn}/measurements/{start}/{to}	Returns the measurements of the node in the requested timeframe
GET	/en/api/v1/nodes/{sn}/measurements/latest	Latest monthly measurements of every sensing service of the node
GET	/en/api/v1/nodes/{sn}/sensing_services Return	s the Sensing Services that are available for the given serial number
GET	/en/api/v1/nodes/{sn}/sensing_services/{sid}/measurements	Returns the measurements of a given service
GET	/en/api/v1/nodes/{sn}/sensing_services/{sid}/measurements/{start}/{	to} Measurements of a service in the requested timeframe
GET	/en/api/v1/nodes/{sn}/actuators	Returns the list of actuators per node that user owns
reque	sts	Show/Hide List Operations Expand Operations Raw
GET	/en/api/v1/requests	Returns API requests count for today
GET	/en/api/v1/requests/{start}/{to}	Returns API requests count for given timeframe

Figure 44: Overview of Synfield API in swagger

2.2.11.1 User authentication and authorization

The first time that the user will try to invoke a web service, the user must type his personal credentials (see Figure 45). Actually, in any web service call, the user's credentials must be used in the HTTP Authorization (Basic type) after the encoding in base64 format. To generate this information, the conversion base64_encode(username:password) must be performed. After that, each call will refer to



user resources. It is worth to stress here that any user is not able to access resources that owns to another user.

external-devices fields act /en/api/v1/fields Implementation Notes Returns the Sections that are of Response Class Model Model Schema	Sign in http://83.235.169.221:9990 Your connection to this site is not private Username	Expand Operations Raw Expand Operations Raw with the matching credentials
ر تفت: ه. Fig	ure 45: Authentication throug	gh user credentials
der /en/api/v1/fields	Returns th	e Sections that are owned by the user with the matching credentials
tesponse Class Addel Model Schema ectionSerializer { nodes (array(NodeSerializer)), name (string), longitude (number): Longitude in degr latitude (number): Latitude in degres, altitude (number): Altitude in meters AddeSerializer { id (integer), node_type (string), node_type (string), name (string), serial_number (string), firmware_version (string), longitude (number): Latitude in degres, altitude (number): Latitude in degres, altitude (number): Latitude in degres, altitude (number): Latitude in degres, altitude (number): Altitude in meters, is_external (boolean): Abstract node th last_data_entry (string); parent_node (string)	ees, ees, iat group external devices;	
lesponse Content Type application/jsc	xn 🔻	
rror Status Codes		
HTTP Status Code Reason		
200 Ok		
400 Bad Request		
481 Unauthorized		
483 Forbidden		
484 Not Found		
429 Too Many Reque	sts	
500 Internal Server E Try it out! Hide Response	rror	

Figure 46: Description of the endpoint that fetches the list of fields of the user

2.2.11.2 Get the list of the fields

The user is able to retrieve the list of fields that owns using the web service as described in Figure 46. Each field includes its name, its coordinates, its time zone and a list of associated Synfield (head and peripheral) nodes (see *Section Serializer*). Each node contains its serial number, its coordinates, its


hardware and firmware version and its name (see *Node Serializer*).Focus on the response model (serializers) and list of the potential HTTP status codes. Figure 47 depicts the response of the web service after the user clicks on "Try it out!".

Request URL	
http://83.235.169.221:9998/en/api/v1/fields	
Response Body	
<pre>{ *nodes": [i *id": 88, *node_trodule": "CSW", *node_type": "Synfield v6.0", *name": "MyField", "serial_number": "ef:12:34:12:ad:33", *firmware_version": "5.0", *longitude": 22.3653, "longitude": 22.3653, "latitude": 39.5168, *altitude": 115:12, *is_external": false, "last_data_entry": null, "parent_mode": null b </pre>	
{ "id": 89, "node_module": "RF", "node_type": "SynRemote_v1",	
Response Code	
200 OK	

Figure 47: List of fields

2.2.11.3 Get the list of sensing services per node

The user is able to retrieve the list of sensing services per node (the services that are measured through the plugged sensors) by given Synfield node's serial number as described in Figure 48. Each sensing service includes its name, its unit, the type of the sensor and the timestamp of the latest measurement. Focus on the response model (serializers) and list of the potential HTTP status codes. Figure 49 depicts the response of the web service after the user clicks on "Try it out!".



GET /en/	api/v1/nodes/{sn}/sensing_services	Returns the Se	ensing Services that are	available for the given serial number
Implementa	ation Notes			
Returns the S	Sensing Services that are available for the g	iven serial number.		
Response C	lass			
Model Mode	el Schema			
Sensingservic id (integer), service_typ sensor (Sen name_el (strinj name_el (st last_change } SensingServic units_change units_el (str units_el (str	revenancer { e (SensingServiceTypeSerializer), sorSerializer), g, tring), ring), ed (string): Timestamp of the latest service mea reTypeSerializer { g, ing), ring)	surement		
SensorSerializ name (string } Response Co	zer { g) ntent Type (application/json 🔻)			
SensorSerializ name (strin) } Response Co Parameters	zer { g ntent Type (application/json 🔻)			
SensorSerializ name (string } Response Co Parameters Parameter	zer { g ntent Type (application/json *) Value	Description	Parameter Type	Data Type
SensorSerializ name (string } Response Co Parameters Parameter sri	zer { g) ntent Type (application/json V) Value (ef: 12:34:12:ad:33)	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (string) Response Co Parameters Parameter sn Error Status	zer { g) ntent Type (application/json) Value [ef: 12:34:12:ad:33] : Codes	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (string) Response Co Parameters Parameter Sn Error Status HTTP Status C	zer { g) ntent Type (application/json Value [ef: 12:34:12:ad:33] ; Codes Code Reason	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (string) Response Co Parameters Parameter Sn Error Status HTTP Status C 200	zer (g) ntent Type (application/json V Value ef: 12:34:12:ad:33 cOdes code Reason Ok	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (strin) Response Co Parameters Parameter Sn Error Status HTTP Status C 200 400	zer (g) ntent Type (application/json) Value ef: 12:34:12:ad:33 i Codes Codes Code Reason Ok Bad Request	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (strin) Response Co Parameters Parameter sn Error Status HTTP Status C 200 400 400	zer { g ntent Type [application/json Value ef: 12:34:12:ad:33] code Reason Ok Bad Request Unauthorized	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (strin) Response Co Parameters Parameter Sn Error Status HTTP Status C 200 400 401 403	zer { g ntent Type application/json Value Value ef: 12:34:12:ad:33 codes Codes Code Reason Ok Bad Request Unauthorized Forbidden	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (string) Response Co Parameters Parameter Sn Error Status HTTP Status C 200 400 401 403 404	zer { g) ntent Type [application/json Value lef: 12:34:12:ad:33] cOdes Codes Code Bad Request Unauthorized Forbidden Not Found	Description Node Serial number	Parameter Type path	Data Type string
SensorSerializ name (string) Response Co Parameters Parameter Sn Error Status HTTP Status C 200 400 401 403 404 429	zer (g) ntent Type (application/json) Value ef: 12:34:12:ad:33 Codes Co	Description Node Serial number	Parameter Type path	Data Type string
SensorSerialia name (string) Response Co Parameters Parameter sn Error Status HTTP Status C 200 400 401 403 404 403 404 429 500	zer { g) ntent Type [application/json ▼ Value [ef:12:34:12:ad:33] cOde Reason Ok Bad Request Unauthorized Forbidden Not Found Too Many Requests Internal Server Error	Description Node Serial number	Parameter Type path	Data Type string

Figure 48: Description of the web service that fetches the list of sensing services of the node







2.2.11.4 Get the list of measurements per sensing service

The user is able to retrieve the list of measurements by given the node's serial number and the sensing service id as described in Figure 50. Each measurement includes the id of the sensing service, its value and the timestamp. Figure 51 depicts the response of the web service after the user clicks on "Try it out!".

GET	/en/api	/v1/nodes/{sn}/sensing_services/{sid}/	measurements	Returns	the measurements of a given service
Impler	nentatio	n Notes			
Return	s the mea	surements of a given service.			
Respor	nse Class				
Model	Model So	hema			
Paginat meas next (previ- count } Measur sensi- value ontim } Respon Param	edMeasur urements (string): UR ous (string t (integer): rementSer ng_service (number), ne (string) ne (string) nse Conter	rementSerializer { (array[MeasurementSerializer]), L to next paginated results,): URL to previous paginated results, Total measurements count ializer { e_id (integer): FK to SensingService, nt Type [application/json]▼			
Parame	eter	Value	Description	Parameter Type	Data Type
sn		ef:12:34:12:ad:33	Node Serial number	patn	string
sid		[671]	Sensing Service ID	path	integer
Error S	Status Co	des			
HTTP S	tatus Code	Reason			
200		Ok			
400		Bad Request			
401		Unauthorized			
403		Forbidden			
404		Not Found			
429		Too Many Requests			
500		Internal Server Error			
Try it o	out				

Figure 50: Description of the web service that fetches the list of measurements per sensing service



Response Body	
<pre>{ "count": 2, "previous": null, "measurements": [{ "sensing_service_id": 671, "value": 18.9, "ontime": "2018-05-15T22:14:35.2539842" }, { "sensing_service_id": 671, "value": 22.9, "ontime": "2018-05-15T22:13:36.5230732" }], "next": null }</pre>	
Response Code	

Figure 51: List of measurements per sensing service

GET /en/	/api/v1/nodes/{sn}/sensing_services/{sid}/m	easurements/{start}/{to}	Measurements of	a service in the requested timeframe
Implement Measurement	ation Notes Its of a service in the requested timeframe.			
Response (lass			
Model Mod	el Schema			
PaginatedMe measurem next (string previous (s count (inte } Measuremen sensing_se	asurementSerializer { ents (array(MeasurementSerializer)),): URL to next paginated results, tring): URL to previous paginated results, ger): Total measurements count tSerializer { rvice_id (integer): FK to SensingService,			
value (num	ber),			
ontime (str	ing)			
}				
Response Co	otent Type application/ison			
D	ment type (appression (see			
Parameter	Value	Description	Parameter Type	Data Type
sn	ef: 12:34:12:ad:33	Node Serial number	path	strine
sid	671	Sensing Service ID	path	integer
start	2018-03-27712-30-007	Valid ISO 8601 date (xxxx-	path	string
	2010-00-21112-30-002	mm-dd) or datetime (yyyy-		sting
	2040.05 40740.00.007	Valid ISO 9601 date (seen	anth	
	2018-00-16116:30:002	mm-dd) or datetime (yyyy- mm-ddThh:mm:ssZ)	parri	string
page	[1	Page number	query	integer
page_size	50	Maximum page size	query	integer
Error Statu	Codes			
HTTP Status	Code Reason			
200	Ok			
400	Bad Request			
401	Unauthorized			
403	Forbidden			
484	Not Found			
429	Too Many Requests			
500	Internal Cap or Error			
	Internal server critor			

Figure 52: Description of the web service that fetches the list of measurements per sensing service in date range



2.2.11.5 Get the list of measurements per sensing service in date range

The user can retrieve the list of measurements by given the node's serial number and the sensing service id in the requested date range as described in Figure 52. The structure of the response is descripted through the serializers. This web service provides also pagination query parameters (maximum page size and number of page). Figure 53 depicts the response of the web service after the user clicks on "Try it out!".

Request URL	
http://83.235.169.221:9990/en/api/v1/nodes/ef%3A12%3A34%3A12%3Aad%3A33/sensing_services/671/measurements/2018-03-27T12%3A30%3A	002/
	•
Response Body	
£	
"count": 2,	
"previous": null,	
"measurements": [
{	
"sensing_service_id": 671,	
"value": 22.9,	
"ontime": "2018-05-15T22:13:36.523073Z"	
33	
"sensing_service_id": 671,	
"value": 18.9,	
"ontime": "2018-05-15T22:14:35.253984Z"	
3	
L.	
"next": null	
) j	
Response Code	
200 OK	

Figure 53: List of measurements per sensing service in date range

2.2.11.6 Get the list of measurement per node in date range

The user can retrieve the list of measurements by given the node's serial number in the requested date range as described in Figure 54. The structure of the response is descripted through the serializers. This web service provides also pagination query parameters (maximum page size and number of page). Figure 55 depicts the response of the web service after the user clicks on "Try it out!".



Figure 54: Description of the web service that fetches the list of measurements per node in date range

diatomic



Request URL

onse Body	
measurements": [
1	
"sensing_service_id": 668,	
"value": 27.1,	
"ontime": "2018-05-15Y22:13:36.523073Z"	
Ъ	
1	
"sensing_service_id": 665,	
"value": 179.55,	
"ontime": "2018-05-15722:13:36.5230732"	
3,	
"sensing_service_id": 668,	
"value": 17.1,	
"ontime": "2018-05-15T22:14:35.2539842"	
h	
1	
"sensing_service_id": 665,	
	•
ionse Code	

Figure 55: List of measurements per node in date range



2.2.11.7 Get the latest measurement of each sensing service per node

The user can retrieve the latest measurement of each sensing service that are associated with the given node as described in Figure 56. The structure of the response is descripted through the serializers. Figure 57 depicts the response of the web service after the user clicks on "Try it out!".

GET	/en/ap	/v1/nodes/{sn}/measurements/latest		Latest monthly measurement	s of every sensing service of the node
Imple Latest	mentatio monthly i	n Notes neasurements of every sensing service of the no	de.		
Respo Model	nse Clas Model S	s chema			
Measur sensi value ontin }	rementSe ing_servic e (number) ne (string)	rializer { e_id (integer): FK to SensingService, ,			
Respor	nse Conte	nt Type application/json 🔻			
Param Param	eters	Value	Description	Parameter Type	Data Type
sn		ef:12:34:12:ad:33	Node Serial numb	er path	string
Error S	Status Co	odes			
HTTP S	Status Code	Reason			
200		Ok			
400		Bad Request			
401		Unauthorized			
403		Forbidden			
404		Not Found			
429		Too Many Requests			
500		Internal Server Error			
Try it	out! Hid	e Response			

Figure 56: Description of the web service that fetches latest measurement of each sensing service per node



Request URL

http://83.235.169.221:9990/en/api/v1/nodes/ef%3A12%3A34%3A12%3Aad%3A33/measurements/latest

Response Body

[
	{
	"sensing_service_id": 673,
	"value": 0,
	"ontime": "2018-05-16T06:29:18.507363Z"
	},
	{
	"sensing_service_id": 672,
	"value": 56.3,
	"ontime": "2018-05-16T06:29:18.507363Z"
	},
	{
	"sensing_service_id": 671,
	"value": 22.4,
	"ontime": "2018-05-16T06:29:18.507363Z"
	},
	{
	"sensing_service_id": 668,
	"value": 21.1,
	"ontime": "2018-05-16T06:29:18.507363Z"
	1

Response Code

200 OK

Figure 57: List of latest measurements (per sensing service) per node



2.2.11.8 Get the list of actuators per node

The user can retrieve the list of actuators by given the node's serial as described in Figure 58. The structure of the response is descripted through the serializers. Figure 59 depicts the response of the web service after the user clicks on "Try it out!".

GET	/en/api	i/v1/r	nodes/{sn}/actuators		Returns the list	of actuators per node that user owns
Impler Returns	nentatio s the list o	on No of act	ites uators per node that user owns.			
Respo	nse Clas	5				
Model	Model S	chem	a			
Actuato id (int node name name value manu updat on_m last_c	rSerialize eger), (NodeSeri (string), e_en (string), e_el (string (integer): ral (booles ted (booles inutes (in change (st	er { jalizer] g),), The st an): Th an): D teger) ring)), tate of the actuator, ue if actuator is used manually. False for auto op enote if the actuator in node is updated, : Define the minutes that the actuator should be	tion, active - If negative, ignore it,		
}						
NodeSe id (int node, node, name serial firmv longii latitu altitu is_ext last_d parer }	rializer { reger), _module (_type (strii (string), _number vare_vers tude (numb de (numb de (numb ternal (bo lata_entri nt_node (s	string ng), (string ion (st iber): La er): La er): Al olean) y (strir tring)), rring), Longitude in degrees, titude in degrees, titude in meters, I: Abstract node that group external devices, rg): utc,			
Respon	ise Conte	nt ly	pe application/json			
Param	eters	Value		Description	Parameter Tupe	Data Tura
sn	eter	Irea	= uired\	Node serial number	nath	string
Error S	itatus Code	des	Reason			
200			Ok			
400			Bad Request			
401			Unauthorized			
403			Forbidden			
404			Not Found			
429			Too Many Requests			
500			Internal Server Error			
Try it o	out!					

Figure 58: Description of the web service that fetches the list of actuators in a node



Request URL

onse Body	
"id": 10,	
"node": {	
"id": 88,	
"node_module": "GSM",	
<pre>"node_type": "Synfield v6.0",</pre>	
"name": "MyField",	
"serial_number": "ef:12:34:12:ad:33",	
"firmware_version": "3.0",	
"longitude": 22.3653,	
"latitude": 39.5168,	
"altitude": 115.12,	
"is_external": false,	
"last_data_entry": "2018-05-16T06:29:18.507363Z",	
"parent_node": null	
},	
"name": "Solenoid Valve (Latching)",	
<pre>"name_en": "Solenoid Valve (Latching)",</pre>	
"name_el": "Ηλεκτροβάνα (εναλλαγή κατάστασης)",	
"value", 0	
contra Contra	

Figure 59: List of actuators per node

2.2.11.9 Update the mode of the actuator

By default, a new actuator is working on manual mode (the user is able to set its state directly from the GUI). The user is able to set the actuator in automatic mode by given its id as described in Figure 60. The automatic mode means that the actuator's state depends on the defined rules in the Synfield platform. No payload is required. The structure of the response is descripted through the serializers. Figure 61 depicts the response of the web service after the user clicks on "Try it out!".

Another web service is available that swaps the mode of an actuator from automatic to manual one. Its description is depicted in Figure 62 while the response after its call by given actuator id is depicted in Figure 63.



PUT	/en/api/	/1/actuators/{actuator_id}/automatic		Upo	dates an Actuator to Automatic Mode
Imple Update	mentation es an Actua	Notes tor to Automatic Mode.			
<mark>Respo</mark> Model	nse Class Model Scl	iema			
Actuato statu }	orAutomati us (string, op	cUpdatePutResponse { tional)			
Respor	nse Conten	Type application/json •			
Param	neters				
Param	ieter V	/alue	Description	Parameter Type	Data Type
actua	tor_id	10	Actuator ID	path	integer
actuar	tor_id Status Coc	10 es	Actuator ID	path	integer
actuat Error S HTTP S	tor_id Status Coc Status Code	10 es Reason	Actuator ID	path	integer
actuar Error S HTTP S 200	tor_id Status Coc Status Code	10 es Reason Ok	Actuator ID	path	integer
Actuar Error S HTTP S 200 400	tor_id Status Coc	10 es Reason Ok Bad Request	Actuator ID	path	integer
actuar Error 9 HTTP 9 200 400 401	tor_id Status Coc Status Code	10 es Reason Ok Bad Request Unauthorized	Actuator ID	path	integer
actuar Error 9 200 400 401 403	tor_id Status Coc Status Code	10 es Reason Ok Bad Request Unauthorized Forbidden	Actuator ID	path	integer
actuar Error S 200 400 401 403 404	tor_id Status Coce	10 es Reason Ok Bad Request Unauthorized Forbidden Not Found	Actuator ID	path	integer
actuar Error 9 200 400 401 403 404 429	tor_id Status Coce	10 es Reason Ok Bad Request Unauthorized Forbidden Not Found Too Many Requests	Actuator ID	path	integer
actua: Error S 200 400 401 403 404 429 500	tor_id Status Code	10 ES Reason Ok Bad Request Unauthorized Forbidden Not Found Too Many Requests Internal Server Error	Actuator ID	path	integer

Figure 60: Description of the web service that sets the actuator in automatic mode

Request URL
http://83.235.169.221:9990/en/api/v1/actuators/10/automatic
Response Body
<pre>{ "status": "ok" }</pre>
Response Code
200 OK

Figure 61: Set the actuator in automatic mode



рит /en/api/	/1/actuators/{actuator_id}/manual		l	Updates an Actuator to Manual Mode
Implementation Updates an Actua	Notes tor to Manual Mode.			
Response Class Model Model Sc	nema			
ActuatorManualU status (string, op }	odatePutResponse { tional)			
Response Conten	t Type application/json ▼			
Parameters				
Parameter	/alue	Description	Parameter Type	Data Type
actuator_id	10	Actuator ID	path	integer
actuator_id	10	Actuator ID	path	integer
actuator_id Error Status Coo HTTP Status Code	10 les Reason	Actuator ID	path	integer
actuator_id Error Status Coo HTTP Status Code 200	10 les Reason Ok	Actuator ID	path	integer
actuator_id Error Status Coo HTTP Status Code 200 400	10 Reason Ok Bad Request	Actuator ID	path	integer
actuator_id Error Status Coo HTTP Status Code 200 400 401	10 Reason Ok Bad Request Unauthorized	Actuator ID	path	integer
actuator_id Error Status Coo HTTP Status Code 200 400 401 403	10 Reason Ok Bad Request Unauthorized Forbidden	Actuator ID	path	integer
actuator_id Error Status Code 200 400 401 403 404	10 Reason Ok Bad Request Unauthorized Forbidden Not Found	Actuator ID	path	integer
actuator_id Error Status Code HTTP Status Code 200 400 401 403 404 429	10 Reason Ok Bad Request Unauthorized Forbidden Not Found Too Many Requests	Actuator ID	path	integer
actuator_id Error Status Code 200 400 401 403 404 429 500	10 Reason Ok Bad Request Unauthorized Forbidden Not Found Too Many Requests Internal Server Error	Actuator ID	path	integer

Figure 62: Description of the web service that sets the actuator in manual mode

Request URL
http://83.235.169.221:9990/en/api/v1/actuators/10/manual
Response Body
{ "status": "ok" }
Response Code 200 ок

Figure 63: Set the actuator in manual mode

2.2.11.10 Update the state of the actuator

Synfield API includes a web service that sets the state of an actuator. Figure 64 describes how to activate an actuator while its response (after its call by given actuator id) is depicted in Figure 65. It is worth to illustrate here that the state of an actuator can be modified only if the actuator operates in manual mode.



PUT	/en/api	/v1/actuators/{actuator_id}/state/{state}			Updates the Actuator state to on/off
<mark>Impler</mark> Update	mentatio es the Acti	n Notes lator state to on/off.			
Respo Model	nse Class Model S	hema			
Actuato statu }	orStateUp Is (string, c	datePutResponse { ptional)			
Respon	ise Conte	t Type application/json ▼			
Param	eters				
Parame	eter	Value	Description	Parameter Type	Data Type
actuat	tor_id	10	Actuator ID	path	integer
state		1	Actuator State [1-255]	path	integer
Error S	Status Co	des			
HTTP S	tatus Code	Reason			
200		Ok			
400		Bad Request			
401		Unauthorized			
403		Forbidden			
404		Not Found			
429		Too Many Requests			
500		Internal Server Error			
Try it o	out! Hid	<u>e Response</u>			

Figure 64: Description of the web service that activates the actuator (manual mode)

Request URL
http://83.235.169.221:9990/en/api/v1/actuators/10/state/1
Response Body
{ "status": "ok" }
Response Code 200 ок

Figure 65: Activates an actuator

2.2.11.11 Register the measurements of the external devices

The Synfield REST API provides two web services that achieve the integration among the Synfield platform and the external devices such as the Plant-O-Meter (provided from BIOS). Especially, a web service supports the insertion of the external devices measurements and the other provides the retrieval of these measurements. Both consider the type of the external device and the serial number as well. By given the type of external device (e.g. plant-o-meter-biosense) and its serial number (e.g.POM31231212), the measurements of the Plant-O-Meter device can be inserted in the Synfield platform through the specific web service as depicted in Figure 66 and Figure 67.



розт /en/api	/v1/external-devices/{device}/serial-number	/{serial_number}/measurem	ents	
		Register one or more meas	urements related t	to a device by given its serial number
Implementatio Register one or r	n Notes nore measurements related to a device by given	its serial number.		
Response Class	5			
Model Model So	chema			
object				
Response Conter	nt Type application/json 🔻			
Parameters				
Parameter	Value	Description	Parameter Type	Data Type
device	plant-o-meter-biosense		path	string
<pre>serial_number</pre>	POM31231212		path	string
Try it out!				

Figure 66: Description of the web service that allows the registration of metrics by device type and device serial number

POST 🗸	http://83.235.169.221:999)/en/api/v1/external-de	levices/plant-o-meter-biosense/serial-number/POM31231212/measurements
Authorization O	Headers (2) Body •	Pre-request Script	
● form-data ●	x-www-form-urlencoded	● raw ● binary	JSON (application/json) 🗸
1 ← { 2 "red": 3 "green" 4 "blue": 5 "infrar 6 "latitu 7 "longi" 8 "date": 9 } 10 11	"2.5456", ':"1.2345", "4.1357", red":"2.336", jde":"39.0742", tude":"21.8243", "01.05.2018. 13:51:54"		

Figure 67: Register measurements for a specific Plant-O-meter device

The above web service is able to consume both an array of measurements as well as one measurement per request. Currently, since the Synfield platform supports only the Plant-O-Meter, the attributes of the payload are depicted in Table 1.

Name	Туре	Description
red	String	The value of the red colour as provided from the Plant-O-Meter
green	String	The value of the green colour as provided from the Plant-O-Meter
blue	String	The value of the blue colour as provided from the Plant-O-Meter
infrared	String	The value of the infrared colour as provided from the Plant-O-Meter
latitude	String	The latitude of the device
longitude	String	The longitude of the device
date	string	The timestamp of the measurement

Table 1: Description of Plant-O-Meter attributes



The potential error status codes related to the above web service are listed the following table.

Table 2: List of potential HTTP status codes in registration of the device measurements

Name	Туре	Description
201	Created	Successful insertion of measurements
400	Bad Request	Erroneous payload
401	Unauthorized	Invalid credentials
404	Not Found	Unsupported type of device, non-existing serial number
500	Internal Server Error	Unhandled exception

2.2.11.12 Get the measurements of the external devices

The retrieval of the measurements can be performed in similar way. By given the type of external device (plant-o-meter-biosense) and its serial number (POM31231212), the measurements of the Plant-O-Meter device can be retrieved through the Synfield API as depicted in Figure 68.

The response of this web service provides an array of measurements, where each measurement includes the attributes is described in Table 3.

Table 3: Description of Plant-O-Meter attributes in the retrieval of measurements

Name	Туре	Description
red	String	The value of the red colour as provided from the Plant-O-Meter
green	String	The value of the green colour as provided from the Plant-O-Meter
blue	String	The value of the blue colour as provided from the Plant-O-Meter
infrared	String	The value of the infrared colour as provided from the Plant-O-Meter
latitude	String	The latitude of the device
longitude	String	The longitude of the device
ndvi	float	The Normalized Difference Vegetation Index as calculated in the Synfield platform
timestamp	string	The timestamp of the measurement (time zone-aware)

The potential error status codes related to the aforementioned web service are listed in Table 4. The measurements related to external devices can be also retrieved through the standard web services that retrieve the measurements related to the Synfield nodes.

Table 4: List of potential HTTP status codes in retrieval of the device measurements

Name	Туре	Description
200	OK	Successful retrieval of measurements
401	Unauthorized	Invalid credentials
404	Not Found	Unsupported type of device
500	Internal Server Error	Unhandled exception



action Notes Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Retrieve the measurement of the device by given its serial number. Retrieve the	xternal-dev	ices		Show/Hide List Oper	ations Expand Operations Ra
Retrieve the measurements of the device by given its serial number. Retrieve the measurements of the device by given its serial number. Response Class Model Model Schema regolated Parameter S Parameter Value Description Parameter S Parameter POM01231212 path string page_11 Query integer Type out the Resonance texponse Content Type (application)(son *) Parameter S Parameter POM01231212 path string page_12 Query integer Type texponse Content Type (application)(son *) Query integer texponse Dody Integer [GET /en/api	/v1/external-devices/{device}/seri	al-number/{serial_number}/me	asurements	
mplementation Notes letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device by given its serial number. Letrieve the measurements of the device biosense Letrieve the measurements of the device biosense Letrieve the measurements of the device of			Ret	rieve the measurements of	the device by given its serial numb
terieve the measurements of the device by given its serial number. Response Class docid [Model Schema warmeter Value Description Parameter Type [application]son * varameter Parameter Value Description Parameter Type [application]son * Parameter Value Description Parameter Value Query integer integer Typit coll Integer Typit coll Integer Typit coll Integer Typit coll Integer Parameter Value Response Body Parameter 1: 2.865, 10: 12: 12: 542: 1: 542: 1: 12: 12: 12: 12: 12: 12: 12: 12: 12	mplementatio	n Notes			
keponse Class Model Model Schema irray(object) tesponse Content Type application/json • Parameter Vulue Description Parameter Type Data Type device plant-o-meter-biosense path string page_stize 100 auery integer page_stize 100 auery integer page_stize 100 auery integer tesponse Body tesponse Bod	Retrieve the mea	asurements of the device by given its	seria <mark>l n</mark> umber.		
Addel Model Schema trray(object) <pre> tesponse Content Type application/joon</pre>	esponse Class	5			
rrayCobject] tesponse Content Type[applicationijson] tarameter Parameter Value Description Parameter Type Data Type device plant-ometer-biosense path string serial_number POM31231212 path string page 1	Nodel Model S	chema			
esponse Content Type [applicationijion •] Parameter Parameter Parameter Parameter Parameter Parameter Parameter Policity 2012 Parameter Parameter Parameter Parameter Policity 2012 Parameter Parameter Parameter Parameter Policity 2012 Parameter Parameter Policity 2012 Parameter Parameter Parameter Policity 2012 Parameter Parameter Policity 2012 Parameter Parameter Policity 2012 Parameter Parameter Policity 2012 Parameter Policity 2012 Parameter Policity 2012 Parameter Pa	rray[object]				
Arameter Value period part of the second Parameter Value period part Type Data Type device period part of the second serial_number POM31231212 path string page 1 duevy integer page_size 100 duevy integer page_size 100 duevy integer tequest URL http://sl.215.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page-18page. tequest URL http://sl.215.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page-18page. tequest URL http://sl.215.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page-18page. tesponse Body [{ "red": 2.5456, "blue": 4.1357, "green": 1.3363, "laftrade": 2.365, "laftrade": 2.365, "blue": 4.1357, "green": 1.3363, "laftrade": 2.365, "blue": 4.1357, "green": 1.2355, "laftrade": 2.365, "laftrade": 2.365	lesponse Conte	nt Type application/ison V			
Parameter Value Description Parameter Type Data Type device plant-o-meter-biosense path string serial_number POM31231212 path string page 1 query integer Try it out Hide Resonne tequest URL http://83.235.169.221:9999/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page=18page= tequest URL http://83.235.169.221:9999/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page=18page= tequest URL http://83.235.169.221:9999/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page=18page= tesponse Body [{ "med": 2.5466, "blue": 4.1357, "infrared": 2.366, "longitude": 21.8243, "infrared": 2.366, "longitude": 21.8243, "infrared": 2.366, "blue": 4.1357, "remers: 1.2345, "infrared": 2.2366, "blue": 4.1357, "remers: 1.2345, "infrared": 2.2366, "blue": 4.1357, "remers: 1.2345, "infrared": 2.2366, "blue": 4.1357, "remers: 1.2345, "infrared": 2.2366, "blue": 4.1357, "remers: 1.2355, "infrared": 2.2366, "blue": 4.1357, "timestamp": "colle=0.25710:59:342" "timestamp": "colle=0.25710:59:342" "timestamp: "colle=0.25710:59:342" "timestamp: "colle=0.25710:59:342" "timestamp: "colle=0.25710:59:342" "time					
radmiteri value va	arameters	Maker	Description	Parameter Tana	Data Tura
serial_number path string page 1 query integer page_size 100 query integer page_size 100 query integer teguestURL tegenestURL tesponse Body tesponse Body [device	plant-o-meter-biosense	Description	path	string
page 1 00 0 000000000000000000000000000000	serial_number	POM31231212		path	string
page_size 100 query integer Try it outi Hde Response tequest URL http://83.235.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page-18page_sex tesponse Body [page	1		query	integer
Try it out Hide Ressonse kequest URL http://83.235.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POM31231212/measurements?page-1&page. kesponse Body [page_size	100		query	integer
<pre>tequest URL http://83.235.169.221:9998/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POH31231211/measurements?page=1&page. tesponse Body {</pre>	Try it out	e Response			
<pre>kequest URL http://k3.235.169.221:9990/en/api/v1/external-devices/plant-o-meter-biosense/serial-number/POH31231212/measurements?page=1&page_ kesponse Body</pre>	and a second sec				
<pre>tesponse Body { { {</pre>	http://83.235	.169.221:9990/en/api/v1/external-d	evices/plant-o-meter-biosense/s	erial-number/POM312312	12/measurements?page=1&page_
I { 'red': 2.5456, 'blue': 4.1357, 'green': 1.2345, 'infrared': 2.336, 'ingitude': 21.8243, 'latitude': 39.0742, 'ndvi: '-0.04294, ''timestamp'', '2018-05-0113:51:542'' }, { 'red': 2.3456, 'blue': 4.1357, ''green'': 1.2345, ''infrared': 2.236, ''infrared': 2.236, 'infrared': 2.236, ''infrared': 2.336, <					
I { "red": 2.5456, "blue": 4.1357, "green": 1.2345, "longitude": 21.8243, "latitude": 39.0742, "ndv": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "longitude": 21.8243, "longitude": 21.8243, "longitude": 21.8243, "latitude": 39.0742, "ndv1": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 OK	Response Body	/			
<pre> { "red": 2.5456, "blue": 4.1357, "green": 1.2345, "infrared": 2.336, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02302, "timestamp": "2018-04-25T10:59:342" } 200 OK </pre>	T				
<pre>"red": 2.5456, "blue": 4.1357, "green": 1.2345, "infrared": 2.336, "longitude": 21.8243, "latitude": 39.0742, "ndv1": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndv1": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 OK</pre>	1				
"blue": 4.1357, "green": 1.2345, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 OK	"red": 2.	5456,			
<pre>"green": 1.2345, "infrared": 2.336, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 OK</pre>	"blue": 4	.1357,			
<pre>"infrared": 2.336, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 0K</pre>	"green":	1.2345,			
<pre>"longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "lnfrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" 200 OK</pre>	"infrared	": 2.336,			
"latitude": 39.0742, "ndv1": -0.04294, "timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T18:59:342" 	"longitud	e": 21.8243,			
"ndv1:: -0.04294, "timestamp": "2018-05-01T3:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndv1:: -0.02392, "timestamp": "2018-04-25T10:59:342" 	"latitude	": 39.0742,			
"timestamp": "2018-05-01T13:51:542" }, { "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342"	"ndvi": -	0.04294,			
<pre> // / / / / / / / / / / / / / / / / /</pre>	"timestam	p": "2018-05-01113:51:542"			
<pre> "red": 2.3456, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.8742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" response Code 200 ОК </pre>	10				
"blue": 4.1357, "blue": 4.1357, "green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndv1": -0.02392, "timestamp": "2018-04-25T10:59:342" tesponse Code 200 OK	Tood"+ 7	2456			
"green": 1.2345, "infrared": 2.236, "longitude": 21.8243, "latitude": 39.8742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" Accessonse Code 200 OK	"blue": A	1357			
"infrared": 2.236, "longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" tesponse Code 200 OK	"dreen":	1.2345.			
"longitude": 21.8243, "latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" esponse Code 200 OK	"infrared	": 2.236.			
"latitude": 39.0742, "ndvi": -0.02392, "timestamp": "2018-04-25T10:59:342" tesponse Code 200 OK	"longitud	e": 21.8243.			
"ndv1": -0.02392, "timestamp": "2018-04-25T10:59:342" tesponse Code 200 OK	"latitude	": 39.0742,			
"timestamp": "2018-04-25T10:59:342" Response Code 200 OK	"ndvi": -	0.02392,			
Response Code 200 ok	"timestam	p": "2018-04-25T10:59:34Z"			
200 OK					
200 OK	esponse Code	2			
	200 OK				

2.3 Virtual Fort Knox Cloud Platform

Virtual Fort Knox (VFK) is a federative platform for the manufacturing industry developed by the Fraunhofer Institute for Manufacturing Engineering and Automation (Fraunhofer IPA). It will offer manufacturing companies an IT strategy that is cost-efficient, agile and scalable. Companies will be provided with efficient access to Industry 4.0 software solutions which are independent of manufacturers, to make advances in the digitalisation and optimisation of their production processes. Figure 69 depicts the concept of VFK. It is based on a cell structure and follows the "security-by-design" principle. Each VFK cell is a securely encapsulated environment for service users and service providers.





Figure 69: Virtual Fort Knox Concept

Figure 70 illustrates the VFK architecture. The physical devices are on the show floor level and are called Smart Objects (e.g. equipment or cyber-physical systems (CPS)). Due to the substantial number of communication protocols, a middleware is used for the communication. This middleware is called Manufacturing Services Bus (MSB) and is described in section 2.3.1.



Figure 70: Virtual Fort Knox Architecture

To communicate with IT services running in the cloud these services are also connected the MSB. Following a service-oriented approach, the services can be aggregated to new services that provide new functionalities. Economically relevant will be the opportunity for Independent Service Vendors (ISVs) to offer their services in the VFK marketplace where the end users are able to purchase the services they need. E.g. an equipment manufacturer can offer some special services for its equipment and the customers can purchase the services which they need. From the technical side VFK offers a software development kit (SDK) which is available in all common programming languages. Applications can be hosted in the cloud infrastructure in form of virtual machines and docker containers. Additionally, the platform provides a flexible middleware as abstraction layer between components which allows changes to the flow of information at run-time.



2.3.1 Cell Concept of Virtual Fort Knox

Each organization, using VFK, operates within an encapsulated environment, referred to as a *cell*. These cells can be publicly hosted or run on a local machine in the network infrastructure of the organization, as shown in Figure 71. Data cannot be transferred between cells, unless applications or smart objects are specifically set up to do so (e.g. a bridge interface). Generally, it is advised to use the middleware accompanying VFK to set up communication within each cell. Each organization with its own cell may consist of multiple users. Users can deploy virtual machines or will be able to download preset software from the shop. Such preset software will be available in the centralized shop, which operates like other app-shops. By default, components which are deployed by a user are only visible, and therefore useable, to him. However, they can be made visible to other users within the organization as well.



Figure 71: VFK cell concept with locally hosted cells and publicly hosted cells in the main infrastructure

2.3.2 Middleware - Manufacturing Service Bus

The Manufacturing Service Bus (MSB) enables a fast and low-effort integration of smart objects or IT-Services, because it provides the integration between various communication protocols such as RESTful Web Service or WebSocket API and various communication standards, for instance OPC UA. For this purpose the MSB provides common interfaces which allow the communication between smart objects and IT-services. The communication process is shown in Figure 72. The data are transferred in an encrypted channel. All send data are transformed to a common data format which ensures that all communication participants can communicate with each other. Received data are added to a queue to allow communication between communication partners with different communication cycles. The routing of the data is done using so-called integration flows, which allow the users to flexibly define where data is forwarded to. Integration flows can be defined without programming skills in the web-based user interface of the MSB. Alternatively, a RESTful API is also available for automation purposes.





Figure 72: Communication Process of the Manufacturing Service Bus

2.3.3 Communication pattern

The communication follows the pattern depicted in Figure 73. At the start of the communication, the client registers itself with the appropriate interface (depending on the used communication protocol). When registering, the client sends its self-description, so that the MSB knows who is registering and what capabilities are available and which data can be expected. After registration is done the client can send data by throwing an event that contains the data. To send data to the client the MSB calls the appropriate function on the client with the data as function parameters.



Figure 73: Communication Pattern of the Manufacturing Services Bus

2.3.3.1 Self-description of services

Each service has a self-description describing its characteristics. The structure of the self-description is shown in Figure 74. A service is classified as an Application or as a Smart Object and can be identified by its unique UUID. Data that is send by a service is described as events. Data that is send to the service can be received as function. Functions can be used to trigger capabilities of the service by internally mapping the incoming function to a callback function in the service-specific code. Such a callback function can trigger return events as well.



Figure 74: Self-description of services

Once a component (smart object or application) is registered at the MSB, the MSB can be configured to transfer information to and from the component automatically by manual configuration via a graphical user interface (GUI). A simple example for the communication pattern is shown in Figure 75. To achieve the shown information exchange, three main configuration steps must be completed: selection of the two components, selection of the corresponding event and function and finally mapping of output data of the smart object to the input parameter of the function of the application. The data emitted by the smart object is attached to the event as a JSON string. The data is then mapped to the corresponding input parameters of the function of the smart application and wrapped in a JSON string again.



Figure 75: Exemplary pattern for data transfer of smart object to an application

For WebSocket communication ready-to-use client libraries in several programming languages are available, which developers of smart objects and applications can use to connect their own product to the MSB.

2.3.3.2 Meta data format

The data format of events and functions is shown in next tables. It is based on the OpenAPI Specification 2.0 (fka Swagger Specification) derived from the JSON Schema for programming language independent definitions of data format. The complete OpenAPI specification that is used for the

liatomic



Swagger-UI as well as for the applications JSON definition can be found under https://github.com/OAI/OpenAPI-Specification.

Table 5

Name	Туре	Format	Comments
Integer	integer	int32	signed 32 bits
Long	integer	int64	signed 64 bits
Float	number	Float	
Double	number	double	
String, Short	string		
Byte	string	Byte	
Boolean	boolean		
Date	string	date-time	As defined by date-time - RFC3339

Table 6

Name	Туре	Format	Comments
Array, List, Set	array	<items></items>	

Table 7

Name	Туре	Format	Comments		
Model	object	<properties></properties>			

Table 8

Name	Туре	Format	Comments
Reference	#/definitions/ <model></model>		

2.3.4 RESTful API

2.3.4.1 Registration

The MSB supports plain old REST to communicate with applications and smart objects. While the WebSocket interfaces allow to infer the connection state of connected smart objects and applications, the REST interface does not allow this. The reason for this is the stateless nature of REST interfaces.

The MSB's REST interface can be reached at port 8083, regardless of the cell the MSB resides in. The OpenAPI specification can be found at the same port under the path /swagger-ui.html. That means, that for MSB reachable under the URL msb.vfk.de, the REST API would be reachable under the URL msb.vfk.de:8083, with the API documentation available under msb.vfk.de:8083/swagger-ui.html.

The Swagger-UI is a documentation tool for APIs that provide an OpenAPI specification. The MSB provides such a specification. While the OpenAPI specification should provide enough information on the interface for basic use cases, it does not provide enough information for complex applications. This documentation serves to supplement the Swagger-UI specification.

As shown in Figure 76, the self-description described in Figure 74 has been extended to support the integration of the endpoints of a RESTful application. A RESTful application can be registered to the MSB in two different ways. One possibility is to use the MSB GUI and the other one the REST API of the MSB.

Listing 1: An example for the specification of a complex object

{	"dataObject": {
	"\$ref": "#/definitions/alarm"
	},
	"alarm": {

H2020-761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation





Figure 76: Extended self-description for RESTful API

2.3.4.2 **Registration with REST API**

ļ

}

}

The REST API two endpoints: One for the registration of an application (Figure 77) and the other one for the registration of Smart Objects (Figure 78). The self-description of the application that should be registered must be described as a JSON object. Listing 2 shows such a JSON description of a simple application and is meant to serve as an example. The fields contained in the JSON definition are described in Table 9



POST /rest/	/applicatior	n/register				/application/register
Parameters Parameter	Value			Description	Parameter	Data Type
application	cation (required) Parameter content type: application/json ▼		application	body	<pre>Model Example Value { "configuration": { "configurationUrl": "string", "location": "string", "parameters": {} }, "connectionFormat": "JSON", "connectionState": "N_A", "connectionType": "WEBSOCKET", "endpoint": "string", "lastContact": "2018-03-08T15:54:38.3212" </pre>	
Response Me	essages					
HTTP Status Co	ode Rea	son	Response I	Model		Headers
201	Crea	ated				
401	Una	uthorized				
403	Fork	bidden				
404 Try it out!	Not	Found				



POST	POST /rest/smartobject/register /smartobject/register						
Param	neters				Parameter		
Param	eter	Value		Description	Туре	Data lype	
smart(Object	(required) Parameter content type:	application/json ▼	smartObject	body	<pre>Model Example Value { "configuration": { "configurationUrl": "string", "location": "string", "parameters": {} }, "connectionFormat": "JSON", "connectionType": "WLBSOCKET", "endpoint": "string", "lastContact": "2018-03-08T15:54:38.3511 ▼</pre>	
Respo	nse Me	ssages					
HTTP S	Status Coo	de Reason	Response M	Vodel		Headers	
201		Created					
401		Unauthorized					
403		Forbidden					
404 Try it	out!	Not Found					

Figure 78: REST endpoint to register Smart Object

Listing 2: Sample self-description of REST application as JSON object

1
"@class": "Application",
"uuid": "71f747a8-b12e-476a-bdb7-85c68c59c282",
"name": "System Information",
"description": "Provides information about a remote system",
"token": "auniquestring",
"events": [
{



"@id": 1, "dataFormat": { "dataObject": { "type": "object", "properties": { "system": { "type": "string" }, "name": { "type": "string" } } } }, "description": "Displays live information about a remote system.", "eventId": "live-information", "name": "Live System Information" }], "functions": [{ "@id": 1, "functionId": "/system-information", "name": "System Information", "description": "Provides System information", "dataFormat": {}, "responseEvents": [1] }], "endpoints": [{ "url": "http://www.example.com", "connectionType": "REST", "functions": [{ "httpMethod": "POST", "connectionFormat": "JSON", "function": 1 }] }]

Table 9: Description of fields contained in JSON object of a self-description

@class	Can either be Application or Smart Object, depending on which type of object is described.							
uuid	A unique identifier for the application. A valid identifier can be generated under <u>https://www.uuidgenerator.net/version4</u> .							
name	The name of the application that will be displayed on the MSB GUI.							
description	A textual description of the application							
token	A token that will be entered in the MSB GUI to complete the registration of the application							
events	A JSON description of the events that the application provides to the MSB. The defined events must provide in incremental numerical id, as well as a unique textual event id. The numerical id is used to refer to the event as a response event in a function definition, while the textual id is used to route the events information within integration flows. The data format follows the OpenAPI specification 2.0. The outer object of the data format must be called dataObject and must be of type object. Everything within the dataObject is optional and can be defined by the developer.							



functions	A JSON description of the functions that the application provides to the MSB. The defined functions must provide in incremental numerical id, as well as a unique textual function id. The numerical id is used to refer to the event as a response event in a function definition, while the textual id is used to route the function information within integration flows. Additionally, the functionId is the path that is attached to an endpoint. If a function can be reached under the URL www.example.com/someFunction, the functionId must be /someFunction. The numerical ID is used under endpoints to connect endpoint definitions with functions. The data format follows the OpenAPI specification 2.0 and is completely developer defined.
endpoints	A description of the endpoints under which the application can be accessed by the MSB. A URL and a connection type must be provided. The functions section further describes how the functions can be accessed by the MSB. The function attribute in the specification refers to the @id attribute of a function defined in the outer application description scope.

2.3.4.3 Registration with MSB GUI

A REST Application can be also added using the MSB GUI. Therefore, you must select the APPLICATIONS tab and press the "+" Button in the left corner. In the pop-up you must press "Create App" and the "Manual app creation wizard" as shown in Figure 79 will appear. In Step 1 an UUID is automatically generated and you can enter other basic information like the name and the description of the application.

Manual app cre 1. Basic Information	ation wizard 2. Endpoints	3. Functions	4. Response Events	5. Verification	6, Rnish!
Basic Information about Application UUID;	t the application. eadf2279-0579-4da0-b7d2-2f9c44	la04ac2			
Application name:*	Sample REST application				
Application description:	Sample REST application				
Cancel			Next		Finish

Figure 79: Manual app creation wizard – Step 1: Basic Information

In Step 2 (see Figure 80) the URL of the REST endpoints must be defined.



Manual app creation wiz	ard 2. Endpoints		4. Response Events	5. Verification	6. Finishi 🔸
Add new Endpoints here: URL*	Connection Ty	pe			
https://sample-rest-server.de	REST	*	İ		
♣add Endpoint					
Cancel		Back	Next		Finish

Figure 80: Manual app creation wizard – Step 2: Endpoints

In Step 3 (see Figure 81) for each defined REST endpoint functions can be defined. The path can contain parameters in the form of {parameter1}. This data format of the parameters must be defined in the "Request Schema" as described in section 2.3.3.2. If the function is called via an integration flow the parameter can be mapped from the triggering event. The parameter will be replaced with the value of provided by the event and the REST call will be executed.

	2, Endpoints	3. Functions	4, Response Exercis	2. Vernual on	
https://sample-res	st-server.de				a
Function: Function Name *	Path *	HTTP Method	Connection Format		
GetData	/data/{userId}	GET	▼ JSON	Ψ	
Description					
"userid" : (
"type" : "string" } }					

Figure 81: Manual app creation wizard – Step 3: Functions

The response of the executed REST call will be send as a response event. The response events for the functions can be defined in Step 4. In the "Response Event Schema" you must describe the data format of the data that will be responded by the REST application (see section 2.3.3.2).



, Basic Information	2. Endpoints	3. Functions	4. Response Events	5. Verification	
tions in Endpoint https://	sample-rest-server.de are:				
GetData					
, Response					8
Response: Response Event Name *	Response Event ID *				
Requested Data	RequestedData				
Description					
Response Event Schema:* { "dataObject" : { "type" : "string"					
2					
<u>ב</u>					

Figure 82: Manual app creation wizard – Step 4: Response Events

In the last two steps you can verify your input and finish the wizard. After that the application will automatically appear in the applications list.

2.3.4.4 Send data

Once your application has been registered and verified to the MSB it is ready to receive information via a function call and to send information to the MSB via an event. An event is send as JSON object with the fields described in Table 10.

uuid	UUID of the Service that sends the event.
eventId	Id of the event as defined in the self-description of the Service.
priority	Priority with which the event is to be processed by the MSB
dataObject	JSON object that contains the data of the event.

Table 10: Description of fields contained in JSON object of an event

The event is then sent to the REST endpoint (/rest/data) shown in Figure 83.



POST /rest/c	data			/data
Parameters				
Parameter	Value	Description	Parameter Type	Data Type
incomingData	(required) Parameter content type: application	incomingData ∞n/json ▼	body	<pre>Model Example Value { "dataObject": {}, "eventId": "string", "postDate": "2018-03-08T15:54:38.341Z", "priority": "0", "uuid": "string" }</pre>
Response Mes	ssages			
HTTP Status Cod	le Reason	Response Model		Headers
200	ОК			
201	Created			
401	Unauthorized			
403	Forbidden			
404 Try it out!	Not Found			

Figure 83: REST endpoint to send data

2.3.4.5 Registration and verification of a component

Upon first boot up of a service, the service connects to the MSB and provides a self-description (more on how to achieve this and details on the self-description in section 2.3.3.1). At this point the component is not yet visible to the user, who needs to activate it first. To do so the user has to navigate to the *SMART OBJECTS* or *APPLICATION* tab, depending on the self-description of the component and click the button for new components ("+"), as can be seen in Figure 84. The classification into smart objects or application is based on the self-description, set by the developer. Its purpose is to allow easier distinction for human users and has no further implication beyond that. As a rule of thumb, smart object should contain at least one sensor or actuator.





H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



The user is then asked to verify the component by inserting the security token which is part of the selfdescription and should be provided by the developer. At this stage the user can also decide if the component should only be visible to him or the entire organization (see Figure 85 below). Visibility decides if other user can see the component when they connect to the MSB with their user credentials. If they do not see the component, they cannot set up new information flows including the component. However, they still might be affected, as information flows set up by one user by trigger actions in organization wide visible components. An example might be a component which takes a considerable amount of time to process data and blocks any other request in this state. One user may find the component permanently blocked when setting up his information flow of another component which is invisible to him.

Virtual Fort Knox Platform X		
Administrator	Deriview Functions Events Configurations Endpoints Flows top	T 12 :
DASHBOARD DASHBOARD INTEGRATION FLOWS SMART OBJECTS MOLUCATION	Verify Application Please enter your verification key.	
C GATEWAYS MGAVITORING ADMIN B USERS	Connection Connection Last co Abbrechen	
PERMISSIONS	+ 6 C	

Figure 85: Security Token and Visibility Setting (MSB GUI)

2.3.4.6 Reviewing of registered components

After completion of the registration process, the new component will be listed in the appropriate tab (*SMART OBJECTS* or *APPLICATIONS*) with its self-description. This description includes general information like name and prose description of its general purpose, as well as specific information regarding the outgoing events the component can throw and functions which can be linked to incoming events, as seen in Figure 86. The component is now ready for the modelling of information flows. If the user wishes to delete the component, he can do so at any time by selecting the component in the respective tab, clicking the small garbage bin icon in the bottom left and confirming his decision.



Virtual Fort Knox Platfor X	/==						[3]	-		×
← → C () guimsbiedu	virtualfortknox.de/#lapplicat	on/iD:196615							ቸ 🕸	ł
۵ 🕥	9. Soche 🖉	= 🕢 Appli	cation A							
Benjamin Götz EDU	Application A	Overview	Events Configurations	Indpoints	flows.	Logs				
		UUD	170c3d65-40d3-4f80-bcc9-c4743ae1782	16						
		Isame	Application A							
INTEGRATION FLOWS		Description	I'm Application A							
SMART OBJECTS										
		Concertion Date								
	0	Connection type	CONNECTED							
	T	Last contact	Fri May 05 16:46:31 CEST 2017							
		_			1					
	Vo	ur Smart	Object will show u	n with						
	-1		object will show up	p with						
	a	l events a	nd functions							
	L	1								
A										
() Incents										
Virtual Fort Knoe 0-2017 Fundhalar Vis	+ 🖬 C									

Figure 86: Detailed Information about Component (MSB GUI)

ê 🕥	Q Search 🐨 🐨	Libelium_Device	e_15_Feb		
doek doek	ME Smart Product 1 VEDSOCIET	Overview Events Function	a Configurations Flows	Ligs	
TestOrga	APZ Dashboard Bridge	PROVIDED CONFIGURATIONS			
	DPAK_HandlerMockup	Name	Format	Value	
	esp Lathe Ethernet			Ubelum Piece Detector	
	G FWLMIllingMachine	Platform	(type:suing)		
SMART OBJECTS	C Libellum, Device	Sensor	("type":"string")	new parameter here	
	(i) Literary Device (1) (a)				
	Machine Traffic Light				
	(test_client				
	And the second second				
1.51					
Virtual Fort Knox	+ 8 C =	You have unsaved changes!			Back Save

Figure 87: Detailed View on Component - Configurations Tab



2.3.4.7 Settings configuration for components via MSB

Optionally, the configurations tab, shown in Figure 87, allows the user to configure internal values of a component remotely via the MSB. This feature is optional and has to be set up by the developer in the program code of the component. If no remote configuration is allowed, the tab name is greyed out and inactive. New parameters can be set by change the value on the right and pushing the orange save button on bottom right. If the new value does not correspond to the required format indicated in the middle row, the changes will not take effect. If a component is modified which is currently offline, the changes will not take effect.

The *Flows* tab show all integration flows which the component is currently a member of (see Figure 88). When activating a new component, this tab is empty. Clicking on the arrow on the right lets the user jump directly to the selected flow.

8	Q (samp) 📰 🐨	Machine Traffic Lig	ht		
doak doak	ME Smart Product 1	Overview Events Functions	Configurations Flows Logs		
TestOrga	APZ Dashboard ME Smart Product 1	ASSOCIATED INTEGRATION FLOWS			
DASHBOARD	DPAK_HandlerMockup	Name	Description	Deployed	Jump to Flow
INTEGRATION FLOWS	O esp Lathe Ethernet	SmartButtonAmpel		0	+
S FLOW SUMMARY	FWLMillingMachine	GrenzwertAmpel			+
O DAARTOBJECTS	Libelium_Device	traffic light test			*
	WEBSOCKET	Lathe-Switch-MTL		197 - 197 -	*
APPLICATIONS	WEBSOCIET	Lathe-BeltTension-MTL		8	+
S GATEWAYS	Machine Traffic Light				
	test_client WEBSOCKET				
	-				

Figure 88: Detailed View on Component - List of all associated Flows (MSB GUI)

2.3.4.8 Data routing with integration flows

2.3.4.8.1 General information about integration flows

On the one hand, the middleware approach replaces the otherwise required point-to-point connections between the components and reduce the maintenance effort for the IT-personnel. On the other hand, it takes over the function of the event listener for all components and allows the configuration of this function at run time. This simplifies the adaption process in case changes to existing solutions are required (e.g. replacement of an old component). It also simplifies the implementation of novel solutions, which rely on data or processing capabilities of existing components and can be configured at run-time without shutdown of the entire system.

To understand how to design components for the use with the MSB, it is useful to first understand, how the user configures his solution based on the available components. Figure 89 shows a simple example for such a solution from view of a user who configures two components to communicate. Once the integration process is complete, the left component can send information to the second one for further processing. The user who implements the solution used a building block concept in a graphical user interface to setup this connection. To achieve the desired information flow, three conditions need to be met, which are represented in Figure 90:

- The components are registered with the MSB (self-description is provided) and activated.
- The information flow is modelled in the MSB by the user.
- The information flow is triggered by the first component in the chain at run-time.





Figure 89: Simple Information Flow modelled in the MSB GUI (only part of GUI is shown)



Figure 90: Steps to compete the Exchange of Information

It is important to understand that each component solely communicates with the MSB and does not per se know preceding or subsequent component in the chain of information. This has implications on the design as intrinsic knowhow between components cannot be expected and should therefore be avoided and all contextually required information needs to be available in the self-description or needs to be exchanged in the events.

2.3.4.8.2 Modelling of an Information / Integration Flow

The MSB is technically able to map events from a component onto functions of the same component. However, this capability should not be used in general to keep the load on the MSB low. Besides, the latency of the MSB typically exceeds component-internal communication by a large margin due to the underlying IP-based communication.

2.3.4.8.3 Initial Creation of an Integration Flow

To build a meaningful information flow, at least two separate components are required. In context of the MSB a model for an information flow are called *integration flow*. Figure 91 shows the first step in creation of such a flow in the *INTEGRATION FLOWS* tab which is initiated similarly to the activation of a



component. Once the blank flow is created, it needs to be named, while a description by the user is optional. The modelling can then be initiated by clicking the *Flow Designer* tab (3. in Figure 91).



Figure 91: First Step in Creating a new Integration Flow (MSB GUI)



Figure 92: Drag and Drop of Components to initiate the Creation of a new Integration Flow



2.3.4.8.4 Selection of Components for an Integration Flow

Within the flow designer view, all available components are show on the right side. If the desired component is not shown, the list can be extended to show all smart objects or all applications by clicking on the respective fields. If the component cannot be found, it has not been activated and the steps of section 2.3.4.5need to be competed again. The components, required for the integration flow need to be dragged and dropped onto the main area, as indicated in Figure 92.

After dragging all components into the main area of the GUI, the user should check if all components are positioned in the correct order according to the desired information flow from left (first component in chain) to right (last component in the chain). This is not necessary but advised, as it improves readability.

2.3.4.8.5 Selection of required Events and Functions for each Component

Before connecting the components, the user needs to select the appropriate event and function for components he wishes to connect next. This is done by clicking on the drop-down menu for the component and then clicking the desired event or function as indicated in Figure 93. By default, the first event from the list in the self-description is selected for all components. If a component does not supply events, the first function from the list in the self-description is set as default. Selecting a function manually may result in a second drop down menu to appear next to the previous one as can be seen in Figure 94.

ð 🕜	Q Search	F 7	New Flow			
dpak dpak	B mockup handler and		Properties Flow Designer			
TestOrga	G SmartMatch		Step 1	Step 2	Step 3	Smart Objects
() DASHBOARD	(A)					Applications
						Scalable.APM.app2 60068409-0259 1104-0716 550055440009
						Event Generator 44454/F-199-489-529-686-984095
CON SOMMANY						AFZ GrafanaServ-InfluxDB Adapter
SMART OBJECTS						Scalable_AFM.app1
						Event-Analyzer-delgate
Sateways			test_client e1077585-e01-4285-020e-3352-08005d	APZ GrafanaServ-InfluxDB Adapt& f7861732-bb5c=15-bb6c-566156e1540	Machine Traffic Light 7421ce07.7620-4c18-a052-60527738cf18g	Delgate-MSB-Bridge Cited/26 6649 4:37 455 cited/080405
			ov event4 •	OperationResultStatus •	shine green -	
				Events		
😟 ADMIN				OperationResultStatus		
				CONNECTED		
				UNCONNECTED		
				Functions		
				Store To Dotatose		
				Store To Database Simple		
				Store To Database as Arrays		

Figure 93: Selection of Event or Function from Drop Down Menu

This implies that the selected function may trigger one of the listed events. The user has to select the desired event from this new list, unless the component is the last one in the chain, where the output event is irrelevant. Due to this behaviour of the GUI it is strongly advised to begin the selection of events and functions at the last component in the desired integration flow.



ê 🕥	Q Search P	Ŧ	New Flow				
dpak dpak	B mockup handler and DERLOYED		Properties Row Designer				
TestOrga	G SmartMatch		Step 1	Step 2	Step 3	*	Smart Objects
	4					RVICE	 Applications
						¹²	Scalable_APM.app2 Scalable_290 1105 aT16 55555548000
							Event Generator Hadikot: CIYI-HIH Q201 00800080005
							APZ GrafanaServ-InfluxDB Adapter 1001727 bbic 1175 bbid Sbia158c1580
U SMART OBJECTS							Scalable APM.app1 SSES400 c250 1104 a110 55655448000
							Event-Analyzer-deigate
GATEWAYS			test_client #9072585-#811-4239-1049-1253269762991	APZ GrafanaServ-InfluxDB Adapter 6061737-bifs-4175-bifs-506415041540	Machine Traffic Light 7421ced7-7a20-ac18-e913-6007/1382/98g		Delgate-MSB-Bridge dried/98466-007-458-615070864.5
			Event4	Store To Datal • OperationResi •	shine green +		-
			-	Response Events Operatoriesutstatui	-		



2.3.4.8.6 Linking of Events to Functions

After the selection process the links between the components have to be set up. Links are always initiated from an event towards a function. The user achieves this by clicking on the orange area of a component with the event and dragging the mouse to the component with the function which he wishes to link to. A successful link is indicated by an orange arrow between the two components, where the arrow is directed towards the component which's function should be executed. An example for successful links can be seen in Figure 95.

ê 🕜	9. Search	F 7	New Flow			
dnek dnek	C mockup handler and		Properties Flow Designer			
TestOrga	G SmartMatch		Step 1	Step 2	Step 3	Smart Objects
DASHBOARD						Machine Traffic Light
						sp Lathe Ethernet
S FLOW SUMMARY						DPAK, Handler Mockup Jacki Jeli-east-ckia-tech-skalitatesaz
						APZ Dashboard Bridge HIQ2/Sel11-529-965/082169981aa
						ME Smart Product 1 Grades-to-4-val 6000-405106175023
Arrecations			test client	APZ GrafanaServ-InfluxD8 Adapter	Machine Traffic Light	Chalism_Device 4884031 sets 466 sub -CoeM178se
GATEWAYS			+1072545+01-4239-03:+1392688996	6061733/b05c-4175-b85d-5802150a1540	7421ce67 7820-4c1 6-ext5-0057738c58g	test_client ent2565 ent4259424e 13524900194
			Event4 -	Store To Datal + OperationResu •	shine green	PWLMillingMachine 649964-tockl-43ut-46u4-68eeMdw153e
O ADMIN						Contraction Device_15_Feb 3xx370094404804000117x278x540
						 Applications

Figure 95: Successfully linked Components

2.3.4.8.7 Mapping of Event Data to Function Parameter Inputs

If the selected functions require no input parameters, specific mapping of parameters can be forgone and the flow is ready to be saved. However, in most cases a mapping of data from the event to the input parameters of the corresponding function is required. The user does this by clicking the small orange dot in the middle of each arrow which will result in a similar GUI to Figure 96. On the left, all available data from the event is displayed and on the right all input parameters are shown. The type is displayed for every parameter as well.




Figure 96: Detail View for Mapping Event Data to Input Parameters for a Function

The MSB is capable of simple type casts which require no further specification such as integer to float, integer to string. Most input and all output parameters are specific to the component developers. However, the parameters mentioned in Table 11 are always available for mapping on the side of the event. The mapping is conducted by selecting the input and output parameters which are supposed to be mapped, indicated by an orange background and clicking the button with the two arrows in the middle. Once the mapping is completed it will show up in the bottom half of the screen, were it can also be deleted by clicking the icon with the garbage bin. Once the user has mapped all function input parameters to the respective event output parameters, he can click the orange save button in the bottom right to save his changes. In general double mappings should be avoided, as they can lead to confusion. If a double mapping occurs, the latest mapping, indicated by being lower in the list, takes precedence. The mapping process has to be repeated for all links (all orange arrows in Figure 95), where mapping is required.

uuid	UUID of the component which sends the event.
eventId	ID given to the event by the developer
priority	Priority set by component developer for transfer by the MSB which might be relevant in case of high load. Possible values are: 0-low / 1(default)-medium / 2-high
postDate	Time when event was thrown by the component.
recieveDate	Time when the event was published to the MSB. The distinction is relevant when a component is set up in such a way that it can function autonomously without MSB connection (e.g. in remote regions without WIFI connection. Optionally events can be buffered in this case and published to the MSB once reconnected.

Table 11: List of standard ever	nt properties
---------------------------------	---------------

2.3.4.8.8 Setting Conditions for Data Transfer

In some cases, the information of an event should only be forwarded to a function when specific conditions are met, which were not foreseen by the developers who designed the components. In this

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



case the *Conditions* tab in the mapping view can be used to set conditions. Conditions can only be set for the data associated with the current event. Other conditions, for example including information from the previous event, are not configurable. In some cases, smart design of the integration flow using the branching (section 2.3.4.8.9) and merging (section 2.3.4.8.10) can be used to achieve the desired results in combination with conditions. A new condition is set by clicking the relevant input parameter form the associated event and selecting the parameter on the left side and clicking the large button in the middle. This yields a view with a drop-down menu like Figure 97 from which the desired comparator can be chosen. Once the user does so, he can set the compare value in a newly appeared field.

uuid : [string] eventId : [string]		/dataObject/message [string]
priority : [integer, int32]		
receiveDate : [string, date-time]		>=
dataObiect : [DBResult]		<=
DBResult : [object]		>
message : [string]	24	1=
result : [boolean]		
operation : [string]		

Figure 97: View to set Condition which incoming Events are forwarded to the next Component in the Flow.

2.3.4.8.9 Branching of Integration Flows

The length of integration flows, as in the number of event to function links, is largely unlimited. The MSB allows more complex designs as well, beside strictly linear integration flows. It is possible to map one event to several functions of one or more components, effectively creating branches in the flow, as shown in Figure 98. In case a subset of these functions is from the same component, the component needs to be dragged and dropped once for each individual mapping. If the branches do not merge again, the user should decide if two separate integration flows (with two separate names) would improve the overall overview.



ê 🕜	A Search P	🔻 😁 Branching Flow			
dpak dpak	mockup handler and interview	Properties Flow Designer			
TestOrga	G SmartMatch	Step 1	Step 2	51ep 3	Smart Objects
					Machine Traffic Light
CONTRABATION PLOWS					Contract and the Ethernet
B FLOW SUMMARY					DPAK_HandlerMockup 2011288 6689 4586 0x31 55953442662
SMART OBJECTS					APZ Dashboard Bridge H400764311-420-940700016900169
					ME Smart Product 1 6810886-1965464158064570617023
APPLICATIONS		test_client	Machine Traffic Light 7421/e07/3/20-4/18/e99-605/7/38458g		Chelium_Device 4886/59-sera-4481 +ush-4564/1995c
GATEWAYS		#3072580-#81-4239-620#-33526880099 	shine yellow •	+	test, client +072583+001-0230-bite=1352sb9096d
		Evente	•		EWLMillingMachine (sut996-3c33-c241 8844-386490191958
O ADMIN			1 100		Delium Device 15, Feb
			Libelium_Device		Applications
			Function3 • Event4 •		

Figure 98: Branching Integration Flow where one Event is forwarded to Two Components

2.3.4.8.10 Merging of Branches in Integration Flows

The merging branches works in a similar fashion as linking and mapping (see Figure 99 for the final view after linking). Once the data of the first event is mapped to the input parameters and saved by clicking the button, the mapping of the data of the second input event can be conducted accordingly. The mappings of the other event will appear in the list of existing mappings (lower half of Figure 96) and vice versa.

ô 🕜	Q _{ibearch}	÷ 🕲	Branch Merging			
dpak dpak.	mockup handler and	Pro	perties Flow Designer			
TestOrga	G SmartMatch		Step 1	Step 2	Step 3	Smart Objects
DASHBOARD						Machine Traffic Light
						esp Lathe Ethernet 200 2007/05 4295-625 5640 52439456075
						DPAK_HandlerMockup 269/0284209-688-2031-556342060
SMART ORBETS						APZ Dashboard Bridge H42KN-ch1-620 P6F 085 Ab5681 as
			test client			ME Smart Product 1 Sitolise 1945-467 959-95 20170823
			49072585-497 4239-6264 33525898009e		DBAK standardischup	Dibelium_Device COBINIC acco +105 acco +106411110ad
S GATEWAYS		71	Event4 -	AP2 GrafanaServ-InfluxDB Adapter 19061737005c41750050-56965641940	Jacon 205-Dati-Dati-Dati-Lon-Minimatian	test, client ext245 ein 429 size 355 sterme
			ME Smart Product 1	n Dur Store To Dataf • OperationResu •	receive5lotNumber •	FWLMillingMachine Gaf9555acki-Opt State
		1	Match			bibolium Device 15 Feb 3-3700 948 400 9560 117/c19-c44
						Applications

Figure 99: Merging of Branches in an Integration Flow

2.3.4.8.11 Wrap up and Saving the Integration Flow

The final steps before the integration flow can be activated, is saving the current setting and activating the flow. Before doing so, the user should be sure that the steps described before are completed. For better overview they are mentioned here once again:

- Name and description represent the purpose of the integration flow sufficiently.
- All components are in the main area of flow manager (at least one square for each component).
- For each component the correct function / event is selected (lower half of the graphical representation of the component).
- All required links are in place (orange arrows).
- All mappings are set as required for each link.
- All conditions (if required) are set (can only be checked in the respective detail views by clicking the orange nobs in the middle of the arrows and switching to the *Conditions* tab).



If this is the case, the user can click the orange save button on the bottom right of the browser window. As a result, the name of the integration flow in the list of flows on the left half of the screen will change from light gray to dark grey, as seen in Figure 100.



Figure 100: Integration Flow after Saving

By default, the integration flow is still deactivated. To activate it, the user has to hit the toggle button to the right of the name of the integration flow. The status of the flow can also be seen by a quick glance on the colour of the icon to the left of the name. Activated flows are indicated by a green icon, while deactivated ones are indicated by a black icon. The activation of the integration flow in the backend of the MSB takes between 1 s - 10 s.



3 Step by step implementation of test scenario for Application Experiment: Agrifood

This section presents a detailed step-by-step guide for integration with the system, which implements the test scenario for Application Experiment: Agrifood, as described in section 1.2.

3.1 A general overview

The integration flow among the modules that are described in section 2 is depicted in Figure 101. The section below provides further information for each step.



3.1.1 Step 1: Create account in Synfield platform

The creation of the account in Synfield platform is necessary. The procedure is described in section 2.2.1.

3.1.2 Step 2: Upgrade the account

The procedure is described in section 2.2.2. After that the user can register Synfield nodes in the platform and use the Synfield API.

3.1.3 Step 3: Configure the fields in Synfield platform

Follow the guidelines in sections 2.2.3 - 2.2.7 to complete this step. This step is very important since it associates the user with his/her fields and the network of devices.

3.1.4 Step 4: Enable devices to send data

Install the Synfield and the Plant-O-Meter devices in the fields and activate them to upload the measurements in the Synfield platform through the Synfield API. This step is also significant since the sensor measures are stored in the Synfield database. The user is able to compose the plots of his/her



interest as shown in section 2.2.8, create/enable rules as described in the section 2.2.10 or detect the spread of diseases as described in section 2.2.9.

3.1.5 Step 5: Register Synfield Platform as application to Manufacturing Service Bus

To integrate the Synfield Platform into VFK the REST API must be mapped to the MSB. The Synfield Platform is represented in the MSB as an Application. Each REST endpoint of Synfield Platform is mapped to an endpoint and function of the MSB Application. The response of the REST call is represented by a response event. The available endpoints of the Synfield REST API are depicted in Figure 44. The representation in the MSB Application is shown in Figure 102. For instance, the endpoint "/en/api/v1/nodes/{sn}/sensing_services" is mapped to the function "Get Sensing Services". The path parameter "sn" is passed as a parameter in the function call. How to define such a MSB Application is described in section 2.3.4.1. The data format of the response event must be defined as described in section 2.3.3.2.

verview Ever	nts Functions Endpoints Config	gurations Flows Logs		
	CTIONS			
 Get Actuators 		/en/api/v1/actuators		
 Update Actual 	tor to Automatic Mode	/en/api/v1/actuators/{actuator_id}/automatic		
 Update Actual 	tor to Manual Mode	/en/api/v1/actuators/{actuator_id}/manual		
 Update Actual 	tor State	/en/api/v1/actuators/{actuator_id}/state/{state}		
 Get Fields 		/en/api/v1/fields		
Get Measurements in Timeframe		/en/api/v1/nodes/{sn}/measurements/{start}/{to}		
 Get Latest Mo 	nthly Measurements	/en/api/v1/nodes/{sn}/measurements/latest		
Data format:	<pre></pre>			
Response ev <mark>en</mark> ts:				
 Response - Description: Data format: 	Get Sensing Services	resp_/en/api/v1/nodes/{sn}/sensing_services		
 Get Measuren 	+ ServiceType: { }, + dataObject: { } }	/en/api/v1/nodes/{sn}/sensing_services/{sid}/measuremen		
C	20 - K	(

Figure 102: Synfield Platform as Application in MSB

H2020–761809: DIATOMIC D3.2: Farm Monitoring and Control Application Experimentation



After the Application is defined it can be registered to the MSB with a POST request sent to <u>https://msb.vfk.de:8083/rest/smartobject/register</u>. The registration process is described in detail in section2.3.4.1. If the registration has been successful, the HTTP Status Code 201 will be returned and the Application can be verified in the MSB GUI with the token defined in the self-description (as described in section 2.3.3.1. If the POST request returns an HTTP ERROR code or the verification is not possible you should check if the self-description is defined correctly.

POST	https://msb.vfk.de:8083/rest/application/register
Parameters	
Key	Value
smartObject	Self description as JSON.

Figure 103: POST request to register Synfield Platform as application to MSB

3.1.6 Step 6: Request data from Synfield Platform

A small sample application has been developed to show how the integrated Synfield Platform can be used within the MSB. The connection between these two applications is done via an Integration Flow (see Figure 104). The sample application sends the event "*Trigger – Get Sensing Services*" to trigger the function "*Get Sensing Services*" of the Synfield API. If this function is triggered, the proper web service of the Synfield REST API will be invoked and the response of this call will be sent as response event to the MSB. The response event can then be forwarded to the requesting sample application which receives the data through the call of the function "*Receive Sensing Services*". The creation of the integration flow is described in detail in section 2.3.4.8. This is just a very simple scenario to illustrate the integration of Synfield API into VFK. But it can be easily extended to include any additional services (e.g. databases, UIs, Machine Learning, …).



Figure 104: Integration flow to request data from Synfield Platform



4 Annex

4.1 Nitrogen rate experiment in maize

4.1.1 Field trial set up

A commercial field was selected for setting up the N rate experiment. The criterion for the field selection was the lack of fertilizer application within 2018. Three maize varieties were selected belonging to different maturity groups (FAO 300, FAO 400 and FAO 500). The N treatments were installed directly after sowing incorporating Urea in the soil using a wheat planter. The experimental deign contains five treatments (different N rates starting from 0 kg of N ha⁻¹ until 200 kg of N ha⁻¹) replicated three times. Each experimental plot contains 4 rows (2.8 m wide) and is 12 m long.

4.1.2 Distribution of Head and peripheral nodes and soil moisture sensors

The head node of the SynField system will be installed at the borders between Var.1 and Var.2 (Figure 1) approximately in the centre of width of the setup in a location that will not interfere with the farming applications. Similarly, the peripheral node will be installed between Var.2 and Var.3. A moisture sensor will be installed in each replication of each variety (9 sensors in total) in the centre of the width of the setup at the plots receiving fertilization of 200, 0 and 100 kg of N ha⁻¹.

4.1.3 Description of measurement procedure with Plant-O-Meter

The measurements will start during the mid-growing stages of maize (around V8) and will continue until maturity. It is expected that measurements will be done every two weeks to monitor the plant status during different growth and reproductive stages of maize. Only the two middle rows in each plot will be measured and the rest will serve as guard-rows. Plant-O-Meter will be set to automatically measure and save results every second. It will be held at a constant distance from plant canopy and moving along the row to cover the whole row length moving up and down along the two middle rows. In addition to that, the second mode for taking stationary measurements in two concentric circles centred on the nodes will be tested.

4.1.4 Expected results in terms of the influence of nitrogen rate on vegetation and soil moisture content

Detailed mapping of plant vigour and biomass will be performed at different growth and reproductive stages of maize. The data from the two extreme and the median fertilizing plans (0, 100 and 200 kg of N ha-1) will be combined with the soil moisture content data derived from the soil moisture sensors. The plants in the non-fertilized plots are expected to appear less vigorous, paler and smaller compared to the other treatments with increasing intensity towards the high rate. The water uptake should be correlated to the biomass and plant vigour since the largest plants need more water. Therefore, it is expected to extract the relationship between crop vigour and biomass with the soil water availability. The system will be tested under real field conditions providing feedback for possible upgrade.





Figure 105: Nitrogen trial setup



4.2 Smart irrigation experiment

4.2.1 Field trial set up

The experiment will be conducted on a 6 ha maize farm, irrigated using solar pumping system. Field is of dimensions 200 x 300 m will be divided on three equal experimental plots. On each plot, different irrigation strategy will be applied. The first treatment will be "farmer's practice" (FP) which is solely based on farmer's experience. The second treatment will be based on soil moisture sensors readings "sensor-based irrigation" (SBI), and the third one will use SBI plus 20% more water, "sensor-based irrigation with excess water" (SBIE). Adjacent field will be used as a non-irrigated treatment and will receive the same management practices as the other three treatments.

All four plots will be treated with sufficient N rate that enables proper growth and development of maize. Only one maize variety, named Exxupery (Agrimatco) belonging to FAO 560 group, has been planted on whole field.

4.2.2 Head node placement and soil moisture sensor distribution

In the initial phase, the electromagnetic probe EMC-38 was used for soil scanning and detection of various properties, such as mechanical soil composition, compression and mineral composition. Based on the scanned results and the elevation model, the field is split to 4 "management zones" with homogenous characteristics (see Figure 106). Since it is expected that different water treatments and soil properties will generate different biomass and plant vigour that in turn will need different amount of water, a minimum of one soil moisture sensor will be placed per one experimental plot plus management zone. Based on this fact a total of 7 sensors will be used. Two sensors are going to be placed on each irrigated treatment and one on the non-irrigated.

The sensors will be integrated in SynField system and their readings will be delivered via head node to the SynCloud and be visible through SynField web app. The head node will be positioned not to interfere with the farming applications.



Figure 106: Sensors topology



4.2.3 Description of measurement procedure with Plant-O-Meter

With the help of Plant-O-Meter phenological stages of the maize will be monitored including growth stages and reproductive maize growth stages. Obtained results will be in the form of vegetation indices, e.g. NDVI, and will give a clear picture regarding crop condition.

It is expected that measurements will be done every two weeks. Two types of measurements will be conducted. The first one is continuous mode which implies automatic measurement and saving data every second. On this occasion, Plant-O-Meter will be positioned at the constant height from the plant canopy and moved along the row. The second one is concentric circles mode centred at the middle of the field. In this mode point measurements will be done that are positioned on the circles and shifted for 45°.

4.2.4 Expected results in terms of the influence of the irrigation strategies

During the experiment a detailed map of vegetation indices will be created for different growth and reproductive stages of maize. It is expected that different irrigation strategies will result in different vegetation indices for the same growing or reproductive stage of the maize. The performance of each strategy will be validated and assessed against specific indicators, such as water consumption per unit and economic efficiency of irrigation.