International Partnership on Innovation

SAMS - Smart Apiculture Management Services

Deliverable N° 4.3

Evaluation of Responses and Support Services

Work package N° 4 – Decision Support System

Horizon 2020 (H2020-ICT-39-2017)

Project N°780755

This project has received funding from the European Union´s Horizon 2020 research and innovation programme under **grant agreement N° 780755.** The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the EU.

Project information		
Lead partner for the deliverable	Latvia University of Life Sciences and Technologies, Faculty of Information Technologies, Department of Computer Systems.	
Document type	Report	
Dissemination level	Public	
Due date and status of the deliverable	30.04.2020	28.04.2020
Author(s)	Dr. Aleksejs Zacepins, Dr. Vitalijs Komasilovs, Dr. Armands Kviesis, Olvija Komasilova	
Reviewer(s)	Stefanie Schädlich (GIZ), Magdalena Sperl (GIZ), Konstanza Joachim (GIZ); Kristina Grazer (UNIGRA)	

This document is issued by the consortium formed for the implementation of the SAMS project under Grant Agreement N° 780755.

SAMS consortium partners

Logo	Partner name	Short	Country
	Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH (Coordinator)	GIZ	Germany
	University of Kassel	UNIKAS	Germany
	University of Graz (Institute for Biology)	UNIGRA	Austria
	Latvia University of Life Sciences and Technologies	UNILV	Latvia
	ICEADDIS – IT-Consultancy PLC	ICEADDIS	Ethiopia
	Oromia Agricultural Research Institute, Holeta Bee Research Center	HOLETA	Ethiopia
	University Padjadjaran	UNPAD	Indonesia
	Commanditaire Vennootschap (CV.) Primary Indonesia	CV.PI	Indonesia

List of Abbreviations

API	Application Program Interface
CPU	Central Processing Unit
CSV	Comma-Separated Values
DSS	Decision Support System
DW	Data Warehouse
GND	Ground
UI	User Interface
QR code	Quick Response code
PCB	Printed Circuit Board
SD card	Secure Digital card
SPA	Single Page Application
USB	Universal Serial Bus

Summary of the project

SAMS is a service offer for beekeepers that allows active monitoring and remote sensing of bee colonies by an appropriate and adapted ICT solution. This system supports the beekeeper in ensuring bee health and bee productivity, since bees play a key role in the preservation of our ecosystem, the global fight against hunger and in ensuring our existence. The high potentials to foster sustainable development in different sectors of the partner regions are they are often used inefficient.

Three continents - three scenarios

(1) In Europe, consumption and trading of honey products are increasing whereas the production is stagnating. Beside honey production, pollination services are less developed. Nevertheless, within the EU 35% of human food consumption depend directly or indirectly on pollination activities.

(2) In Ethiopia, beekeepers have a limited access to modern beehive equipment and bee management systems. Due to these constraints, the apicultural sector is far behind his potential.

(3) The apiculture sector in Indonesia is developing slowly and beekeeping is not a priority in the governmental program. These aspects lead to a low beekeeper rate, a low rate of professional processing of bee products, support and marketing and a lack of professional interconnection with bee products processing companies.

Based on the User Centered Design the core activities of SAMS include the development of marketable SAMS Business Services, the adaption of a hive monitoring system for local needs and usability as well as the adaption of a Decision Support System (DSS) based on an open source system. As a key factor of success SAMS uses a multi stakeholder approach on an international and national level to foster the involvement and active participation of beekeepers and all relevant stakeholders along the whole value chain of bees.

The aim of SAMS is to:

- enhance international cooperation of ICT and sustainable agriculture between EU and developing countries in pursuit of the EU commitment to the UN Sustainable Development Goal (SDG N°2) “End hunger, achieve food security and improved nutrition and promote sustainable agriculture”
- increases production of bee products
- creates jobs (particularly youths/ women)
- triggers investments and establishes knowledge exchange through networks.

Project objectives

The overall objective of SAMS is to strengthen international cooperation of the EU with developing countries in ICT, concentrating on the field of sustainable agriculture as a vehicle for rural areas. The SAMS Project aims to develop and refine an open source remote sensing technology and user interaction interface to support small-hold beekeepers in managing and monitoring the health and productivity in their own bee colonies. Highlighted will be especially the production of bee products and the strengthening of resilience to environmental factors.

- Specific objectives to achieve the aim:

- Addressing requirements of communities and stakeholder
- Adapted monitoring and support technology
- Bee related partnership and cooperation
- International and interregional knowledge and technology transfer
- Training and behavioral response
- Implementation SAMS Business cooperation

Contents

List of tables/figures

Executive summary

All bee colony monitoring data coming from the SAMS monitoring devices is accessible globally using a web-based application – the SAMS Data Warehouse. This report describes the new web user interface for the SAMS Data Warehouse (DW) and the clickable version of the front-end for mobile devices. As well issues related to the bee colony monitoring data transfer and system installations are addressed. Sustainability aspect of the SAMS data warehouse is described and DW deployment process is summarised. The response and support actions are elaborated for different regional settings in target countries.

1. New version of the SAMS data warehouse

The SAMS Data Warehouse (DW) is a universal system, which can operate with different data inputs, such as direct data transfer from bee colony monitoring devices, manual data upload and has flexible data processing algorithms based on the incoming data sets. The SAMS Data Warehouse is already developed and is a fully operational platform and can easily be accessed from any web browser (except Internet Explorer). The SAMS bee colony monitoring devices, or any other monitoring devices which are available, can directly be connected to the SAMS DW and transfer and store the data. A description of the SAMS DW base and its containing parts (as well as their interaction) is provided in the Report on Data Management (D.4.1.).

The LLU team has developed the basic "Instruction on how to connect general bee colony monitoring hardware to SAMS Data Warehouse". This instruction will ease the process of sending data from any general available bee colony monitoring system to the SAMS data warehouse for storage and analysis. Users can easily sign up to enter the SAMS DW using their own email/ password or google account. Each user is associated with one or more workspaces, which should be configured based on apiary configuration, with their own connected monitoring systems.

Then, the following steps must be performed on the monitoring device (hardware) to send its own data to the warehouse:

- Acquire an access token. This token is used by the DW to authenticate and authorize the request.
- Post/transfer the data to the DW. There is a single endpoint for posting data to the DW – https://sams.science.itf.llu.lv/api/data. The access token should be provided in the Authorization header and the request body can contain multiple data packages.

Reports about posted measurements are immediately available in the user interface (UI) in the *Reports* section. Additional debugging information is available in the dashboard (last 10 received measurements) and devices (last events and errors).

Full instruction on how to connect general bee colony monitoring hardware to the SAMS Data Warehouse is available here:

https://sams-project.eu/wp-content/uploads/2020/02/DW-data-sending-guide.pdf

The user interface of the web version is updated and modified to be more friendly for the end-users. The interface is built to be more in line with the currently developed user interface design of the mobile app. Now it is more compatible with mobile devices. Similar colour schemes and interface elements (buttons, lines, list elements) are used. The SAMS mobile app will be further described in the next section.

In the previous version the menu was located on the top (see Figure 1) while it is now located on the left side of the screen (see Figure 2). This is done to ease the use of the menu on the mobile devices that usually have smaller size screens:

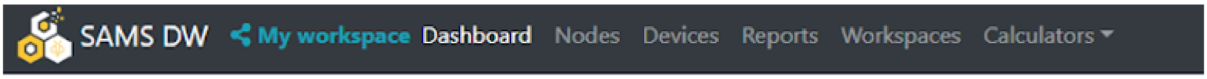


Figure 1. SAMS DW previous menu location

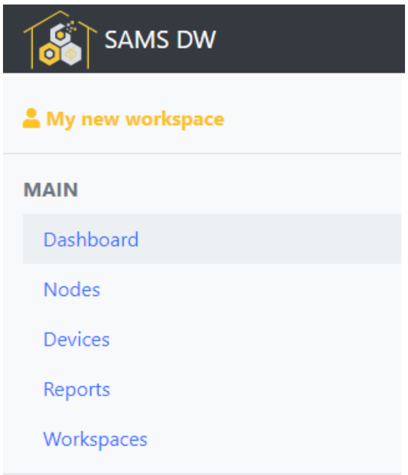


Figure 2. SAMS DW new menu location

Also, the dashboard is fully restructured, and information is shown in a more compact view (see Figure 4):

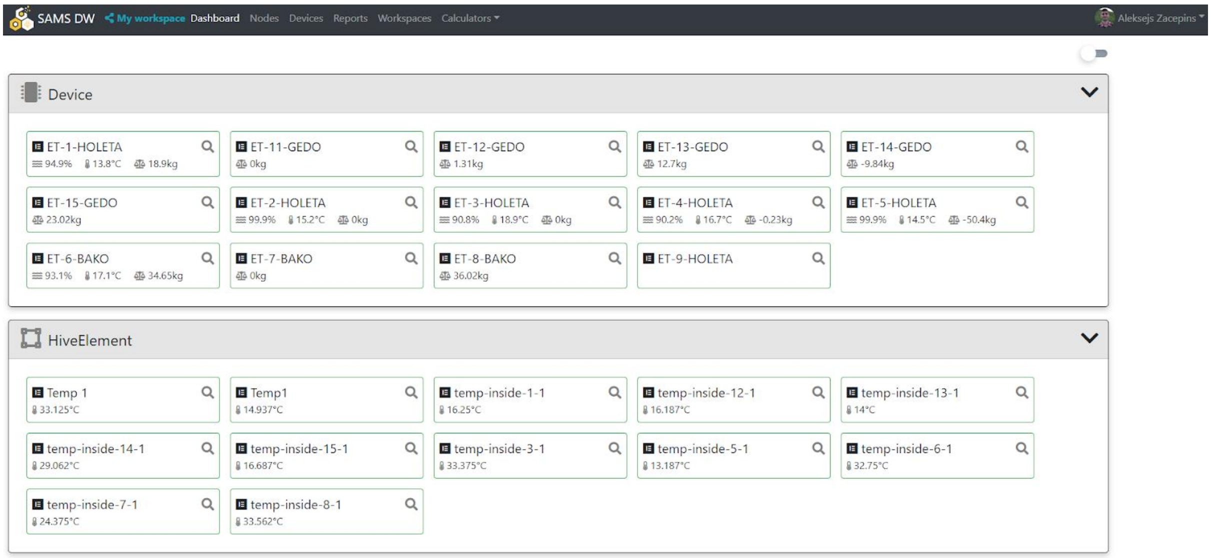


Figure 3. Dashboard view in the previous DW version

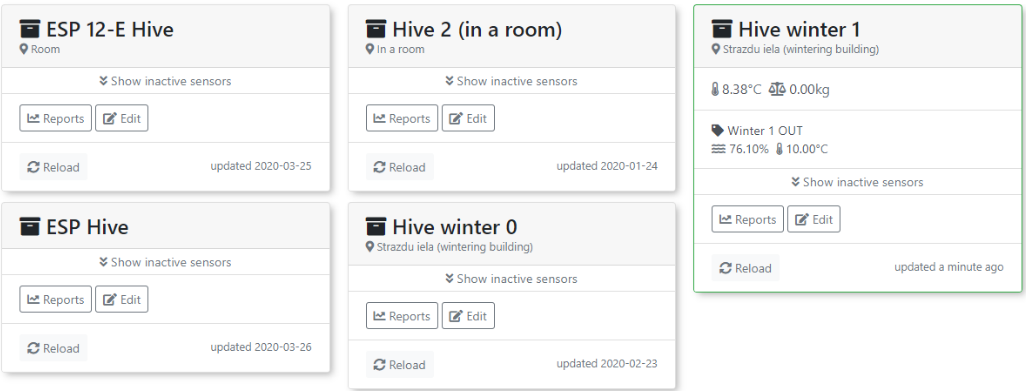


Figure 4. Updated dashboard view in the SAMS DW

In this case, inactive sensor data is hidden by default and in case that the user needs older data, it can easily be shown to the user. A hive is considered inactive if the data is not constantly (at least every twelve hrs.) flowing/ transferred. In comparison active hives have a green thin border, that indicates that monitoring data (at least one sensor is operating and transfers data) is flowing into the DW.

The new button “Reload” was added to give the user the option to renew the last measurements. This is mostly needed for testing purposes, to validate frequent data updates and check the connection between hardware and DW instantly.

The report view has been also updated, to enable users to also see the full screen on a smaller device such as a laptop screen or a smartphone. The measurements are shown, without values, in the chart in the simple *Report view*, but values can be viewed by clicking “Show table” to activate *Report view with measurements.*

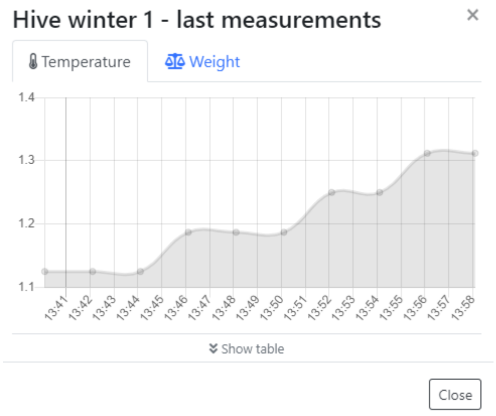
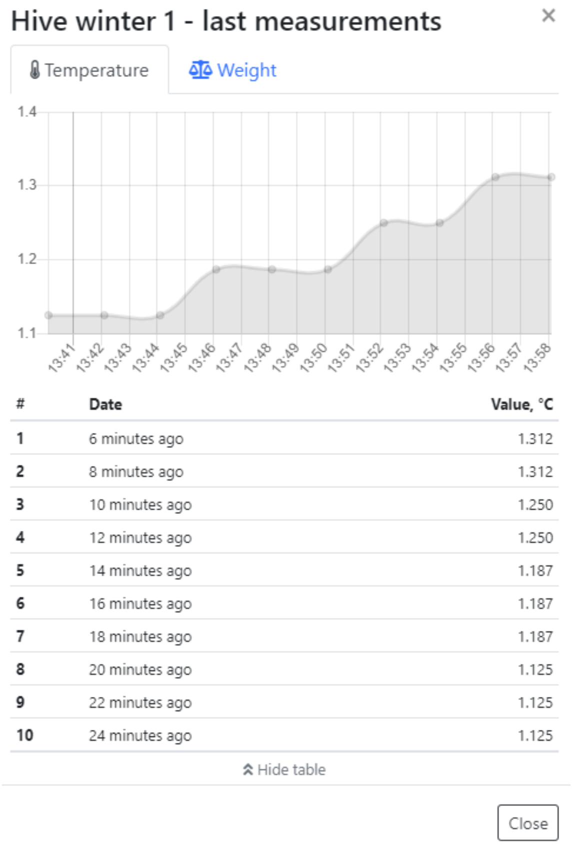


Figure 5. Report view

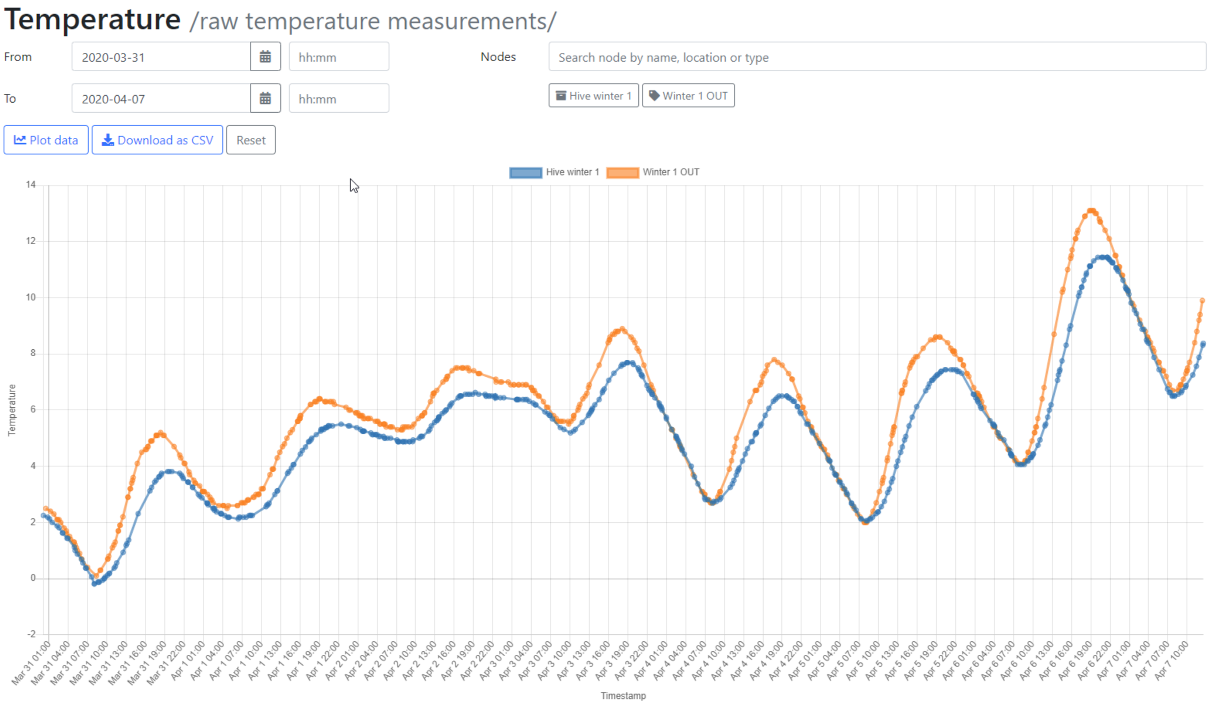


#	Date	Value, °C
1	6 minutes ago	1.312
2	8 minutes ago	1.312
3	10 minutes ago	1.250
4	12 minutes ago	1.250
5	14 minutes ago	1.187
6	16 minutes ago	1.187
7	18 minutes ago	1.187
8	20 minutes ago	1.125
9	22 minutes ago	1.125
10	24 minutes ago	1.125

Figure 6. Report view with last measurements

An additional option to provide an overview on different parameters on one individual chart was implemented to see correlations and dependencies between several parameters. A detailed report view can be activated by clicking button “Reports” in the dashboard.

Furthermore, an export data option is now available in the data warehouse. So, users can download measurements from their own hives by clicking “Download as CSV” in the DW. This feature was mainly implemented for the scientific community to give them the option to analyse the data using different approaches and methods.



Temperature /raw temperature measurements/

From 2020-03-31 hh:mm

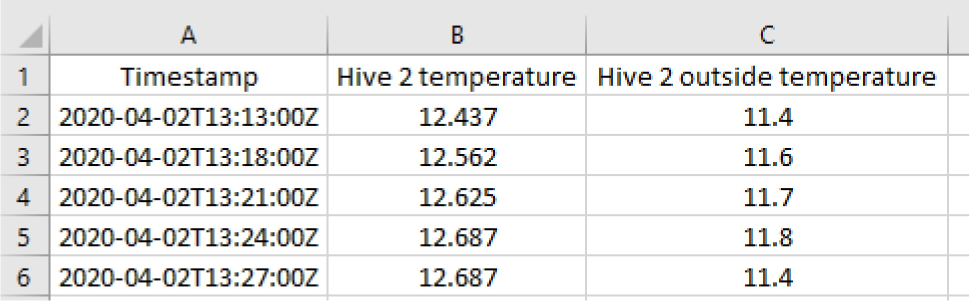
To 2020-04-07 hh:mm

Nodes: Search node by name, location or type

Hive winter 1, Winter 1 OUT

Figure 7. Detailed report view in the SAMS DW

After the file has been downloaded it can be used in any way the user prefers. CSV files have a simple structure in which the first column corresponds to the timestamp when the measurement was taken (shown in UTC time zone). Next columns show the data/ parameters of the specifically chosen colony/ monitoring system.



	A	B	C
1	Timestamp	Hive 2 temperature	Hive 2 outside temperature
2	2020-04-02T13:13:00Z	12.437	11.4
3	2020-04-02T13:18:00Z	12.562	11.6
4	2020-04-02T13:21:00Z	12.625	11.7
5	2020-04-02T13:24:00Z	12.687	11.8
6	2020-04-02T13:27:00Z	12.687	11.4

Figure 8. CSV file generated out of the system

2. Design of the Decision Support System

For the development of the Decision Support System (DSS) SAMS mobile app the SAMS project consortium agreed to focus on the first end-users – the beekeepers.

The clickable version of the user interface for the native mobile devices is developed using the Figma online tool (https://www.figma.com/). Figma helps teams to create, test, and create better designs from the very first step to the final product. The SAMS consortium is not going to implement a final working mobile application for the beekeepers but will leave that as a possibility for the business development process. As the developed design and front-end are available open-source any interested individual will be able to access and implement it, as well make use of it for its own purposes.

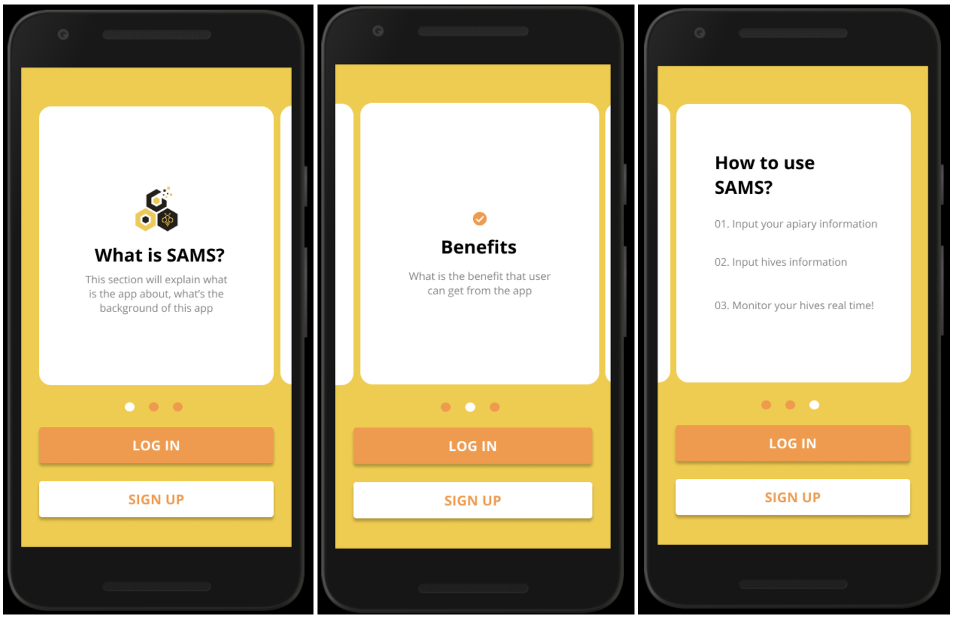
The design of the front-end is developed mainly by CV.PI in close collaboration with our advisory board member Labtek Indie and all SAMS project partners. The end-users (beekeepers) were involved in the development of the technology through a User Centred Design approach. The iterative development process was discussed between CV.PI, UNILV, GIZ and ICEADDIS during regular skype meetings which were especially dedicated to this topic.

Several interviews with individual beekeepers (end users) were conducted by the local project partners to define the specific user needs and the basic requirements for the mobile application interface.

The developed design consists of 37 screens fully adapted to a mobile device, aiming to have the best user experience for the beekeeper with simple, responsive and self-explanatory pages / screens.

Some of the pages do clearly correspond to the front-end elements integrated in the web version (Data Warehouse) of the user interface.

Below all screens are presented and a comparison with the web version is provided.



What is SAMS?

This section will explain what is the app about, what's the background of this app

Benefits

What is the benefit that user can get from the app

How to use SAMS?

01. Input your apiary information

02. Input hives information

03. Monitor your hives real time!

Figure 9. Introduction pages of the mobile app

As usual, the front page provides basic information about the SAMS project itself but also on the background of the app and instructions on how to use the app.

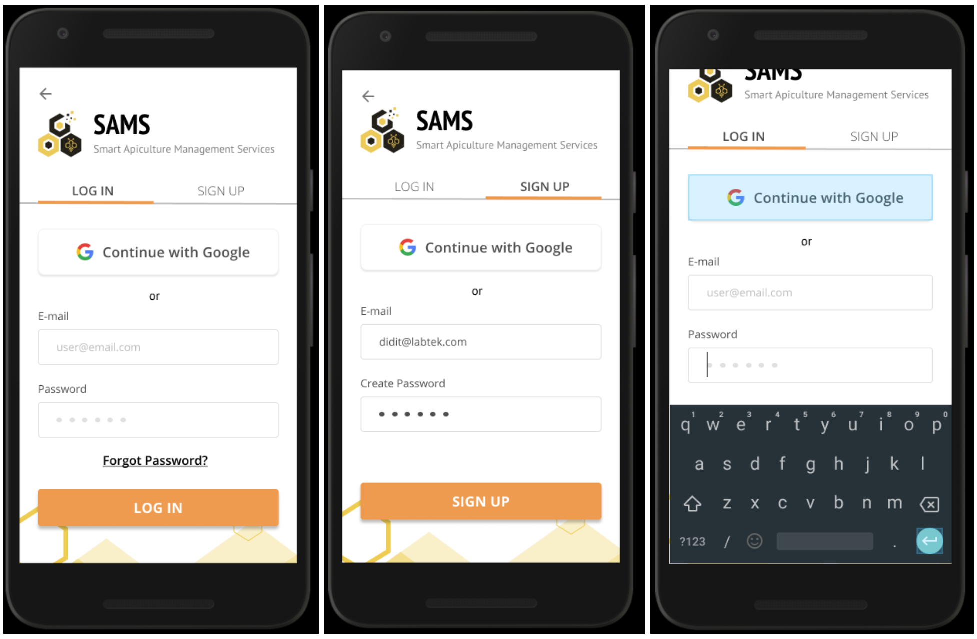


Figure 10. Log in and Sign up screens of the mobile app

The Log in or Sign up button is implemented using the Auth0 service. This approach is chosen as Auth0 since in this way it provides authorization and authentication service, which means in the opposite way that the SAMS project does not have to deal with the user credential management. The user interface for the authorisation is similar to all versions of front-ends as it is provided by Auth0 service.

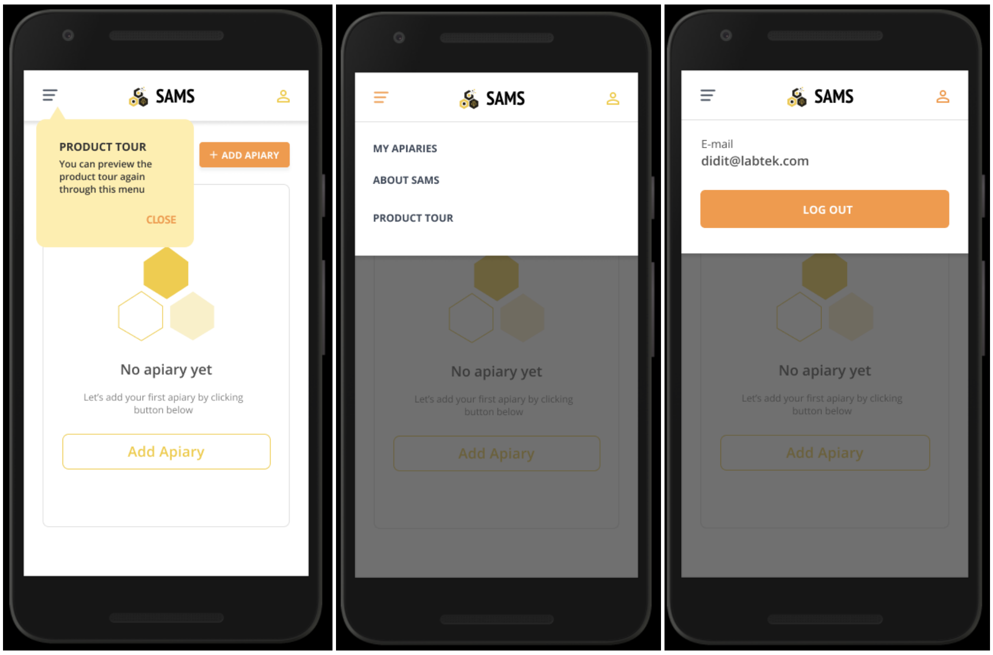
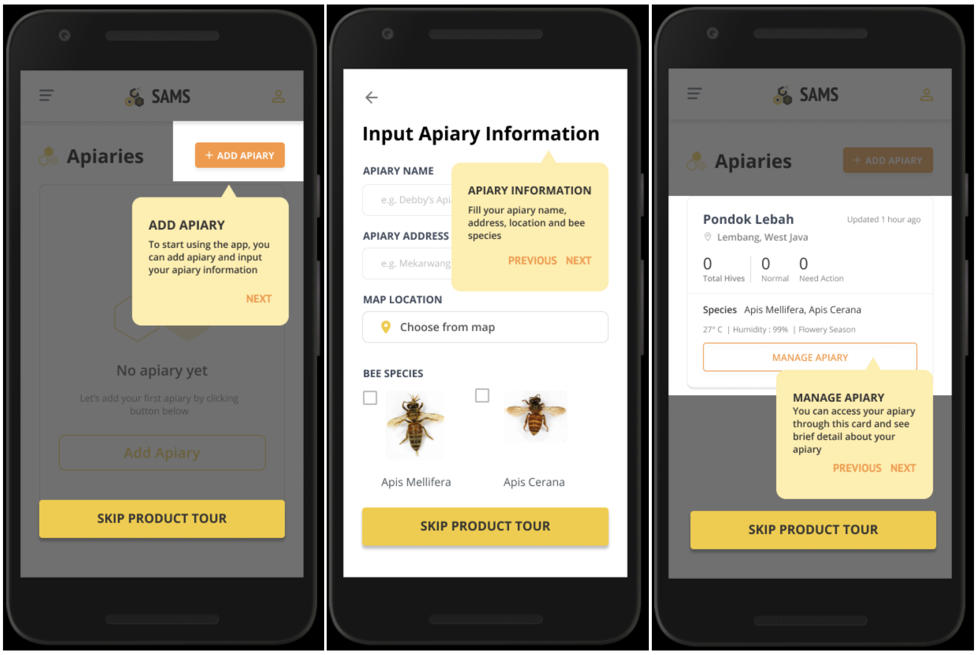

SAMS
PRODUCT TOUR
You can preview the product tour again through this menu
CLOSE
+ ADD APIARY
No apiary yet
Let's add your first apiary by clicking button below
Add Apiary
SAMS
MY APIARIES
ABOUT SAMS
PRODUCT TOUR
No apiary yet
Let's add your first apiary by clicking button below
Add Apiary
SAMS
E-mail
didit@labtek.com
LOG OUT
No apiary yet
Let's add your first apiary by clicking button below
Add Apiary

Figure 11. Menu for starting the app tour and user information

The app menu, provides a list on the beekeepers included apiaries, the SAMS product and enables the user to start an interactive product tour. By clicking the user icon, the users can easily log out from the system.



ADD APIARY

To start using the app, you can add apiary and input your apiary information

APIARY INFORMATION

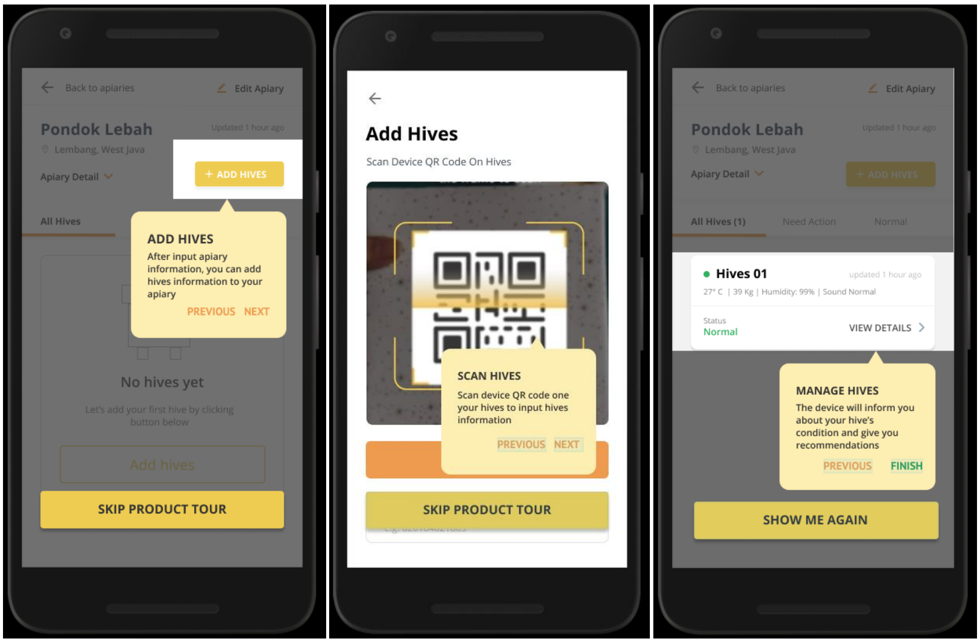
Fill your apiary name, address, location and bee species

MANAGE APIARY

You can access your apiary through this card and see brief detail about your apiary

Figure 12. Apiary information

When a user logs in, the product tour automatically starts and the user can go through the main app functions and functionalities. At the first stage the user must add its apiary and provide basic information about it, including the apiary name, the address, the location (optional input field) and the bee species (if applicable). If applicable, there are currently only three options to choose of - Apis Mellifera and Apis Cerana or both. When an apiary is created, it can be observed and edited.



ADD HIVES

After input apiary information, you can add hives information to your apiary

SCAN HIVES

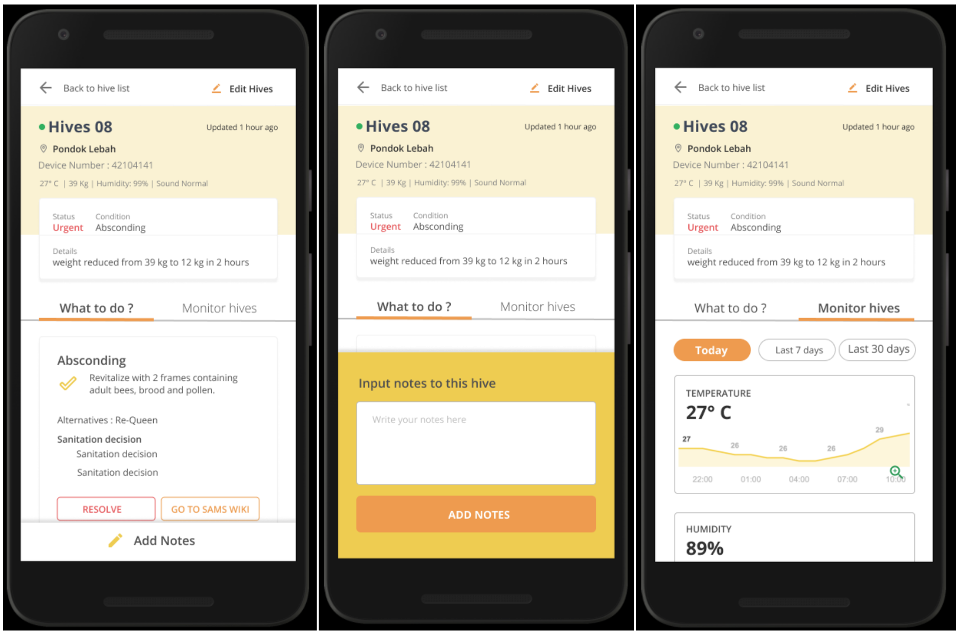
Scan device QR code one your hives to input hives information

MANAGE HIVES

The device will inform you about your hive's condition and give you recommendations

Figure 13. Hive information

The next step is to add the hives. Unlimited number of hives can be added to the apiary. A hive name (e.g. Hive 01, Hive 02) is suggested automatically to the user but the user can also rename the suggested hive name. To ease the process of adding the hive information, QR code scanning can be done. QR code stores pre-configured device information. Moreover, to identify the hives easily, there is the option to upload a picture of each hive. After the hive has been added it can be observed and the app will provide the information about the hive condition and can give recommendations on possible required actions from the user's side.



Hives 08 Updated 1 hour ago
Pondok Lebah
Device Number : 42104141
27° C | 39 Kg | Humidity: 99% | Sound Normal

Status: Urgent
Condition: Absconding
Details: weight reduced from 39 kg to 12 kg in 2 hours

Absconding
Revitalize with 2 frames containing adult bees, brood and pollen.
Alternatives : Re-Queen
Sanitation decision
Sanitation decision
Sanitation decision

Hives 08 Updated 1 hour ago
Pondok Lebah
Device Number : 42104141
27° C | 39 Kg | Humidity: 99% | Sound Normal

Status: Urgent
Condition: Absconding
Details: weight reduced from 39 kg to 12 kg in 2 hours

Input notes to this hive
Write your notes here

Hives 08 Updated 1 hour ago
Pondok Lebah
Device Number : 42104141
27° C | 39 Kg | Humidity: 99% | Sound Normal

Status: Urgent
Condition: Absconding
Details: weight reduced from 39 kg to 12 kg in 2 hours

TEMPERATURE
27° C

HUMIDITY
89%

Figure 14. Detailed hive view

Users can see a detailed view of the hive, its issues and main recommendations to overcome those difficulties. It is also possible to add notes for the hive and observe colony main parameters. In order to resolve detected difficulties of the specific bee hive, the front-end provides the option for the beekeeper to visit the SAMSwiki page for detailed instructions and proper responses.

Next steps are the evaluation of the interface by the end users and to adapt the interface accordingly. Adaptation of the interface to the local needs and languages can be done in future by local enablers and possible start-ups. It is concluded that the developed design / interface will be evaluated by different beekeeper (end-user) groups – first the English interface, then the translated ones in Amharic and Bahasa Indonesia.

The response and support actions will be elaborated for different regional settings.

A good feature of the SAMS technological approach is that the back-end and front-end are separated. This means that different front-end versions can be developed and used with the one developed back-end. For example, in the future different front-ends can be developed for each target country with the inclusion of country specific needs and peculiarities. For example, mostly in West Java, Indonesia, beekeepers keep *Apis Cerana* and they are more interested in the absconding status. Whereas in Ethiopia, where the majority of the bees kept is *Apis Mellifera*, information on bee swarming is more interesting for beekeepers. It is just one example of how the collected data can be used differently and accordingly to the different settings/ contexts and the specific/ individual end users' needs. The actual collected data remains the same, but which data is shown on the screen is highly depending on the beekeepers and their request of the different region.

The before mentioned SAMSwiki was developed as knowledge base about bee aspects. SAMSwiki is a resource where also management practices and decisions are described, its usage is described in the report Bee-management and bee-health database (D5.2). SAMSwiki is directly linked to the SAMS DSS and is used as a source for gaining required information to automatically identify specific bee colony state. SAMSwiki is available in different languages (so far in English, Amharic, Bahasa Indonesia) to reach out to a wider audience in target countries. For example if a colony has absconded, beekeepers will be directed to the SAMSwiki page, where they can get further information on the absconding event (a required further actions after this particular colony event occurred).

3. Problems related to the data transfer and monitoring system installations

The idea and purpose of the SAMS DW is to store bee colony data, that has been collected by the bee colony monitoring devices. Within the SAMS project bee colony monitoring primarily will be done in Indonesia and Ethiopia. In Latvia and Germany hardware is installed for system testing and debugging purposes. Systems installed in Latvia and Germany are working properly, which means that the connection between hardware and SAMS data warehouse is operating well.

Nevertheless, data collection and transfer from the target countries to the SAMS DW is not happening as it was expected due to different reasons in the countries. Such reasons are:

- One of the problems in Ethiopia was related to 3G mobile network coverage. The provided router for the system cannot send data due to mobile network problems (possibility that the router could not switch to a reliable network, 2G in this case). It was agreed that a router with capabilities to use 2G network for data transmission will be used instead. Installation process was planned for March 2020 in Ethiopia and April 2020 in Indonesia. Due to the Corona Virus it was not possible to do that and is postponed until further notice and restrictions are getting less.
- Due to challenges in maintaining stable connectivity and a lack of IT expertise near the installation sites of Holeta and Gedo in Ethiopia for applying a quick solution/ fixing the problems systems are not operating.
- The biggest issue in Indonesia is related to the absconding of bees. Many bee colonies of our partner absconded due to the dry season that lasted until the end of 2019 in Indonesia.
- One of the challenges for both countries is the mobile network operation for data transmission. Network stability issues and frequent interruptions have been observed.

It is concluded that the option for local storage of measurements is necessary to mitigate the data transmission problems. It is clear that in this case real-time decisions could not be made, but historical behaviour of the colony would be available.

Unfortunately, until today there was no constant monitoring system available in Ethiopia and Indonesia.

The partners from the LLU developed a solution for convenient offline data transfer from local bee colony monitoring device to the flash drive.

In general, the proposed solution involves several steps which the user (beekeeper/local engineer/ system's owner etc.) needs to take to copy the local bee colony monitoring data to the removable USB drive. The monitoring system's main board (Raspberry Pi Zero W) (see Figure 15) itself needs to be equipped with an additional push-button (signal the Raspberry Pi to start file transfer), a status LED (show that file transfer is in progress) and a USB OTG (On-the-go) cable that acts as an adapter for micro-USB to USB. To make this solution work for the SAMS monitoring devices (that already have a printed circuit board (PCB), therefore adding additional components would require one to redesign the PCB) already existing components on the Raspberry Pi Zero W can be utilized. For example, to show status of the file transfer process the already existing activity LED (*act_led*) on the Raspberry Pi Zero W (normally shows the CPU/SD card's activity) can be used, because it is possible to take over the control of the *act_led* (hence no additional LED is needed to attach to the Raspberry Pi Zero W). To start the file transfer process user should somehow "inform"/ signal the Raspberry Pi Zero W that it should now copy the files. This can also be simulated by simply connecting the input pin to GND (e.g. with a jumper wire or jumper shunt).

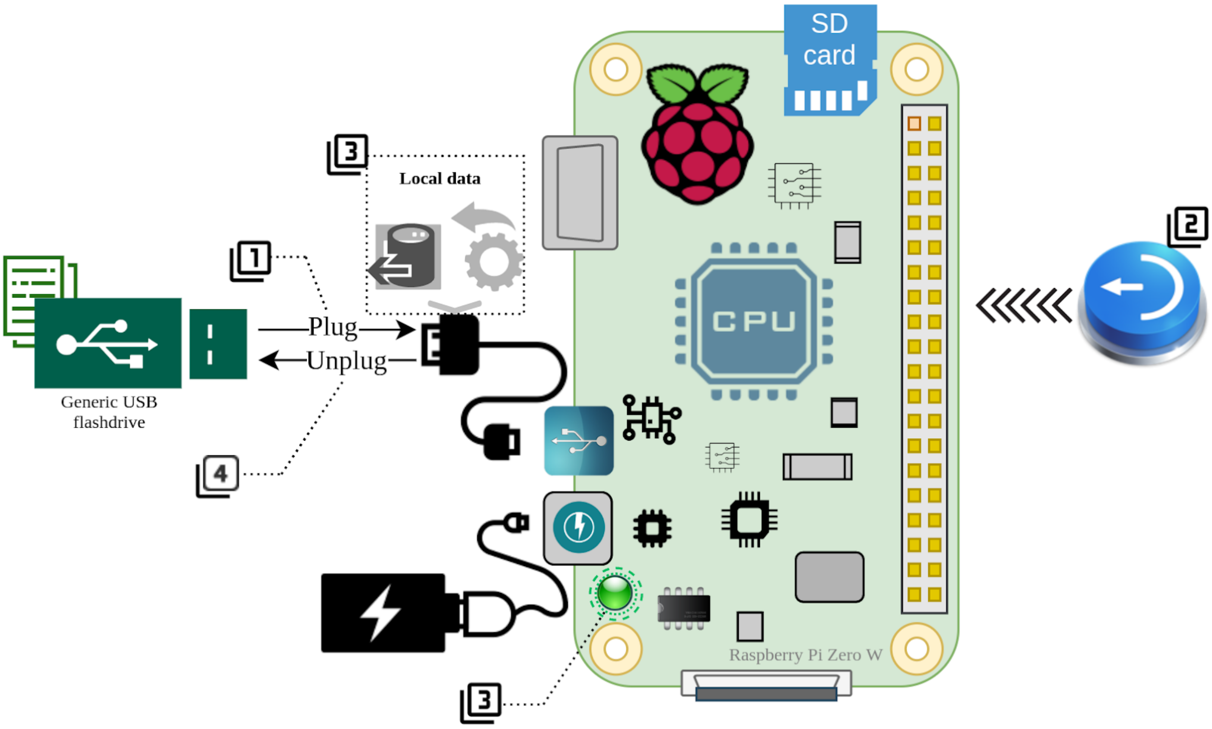


Figure 15. Concept for the local data retrieval

Regarding the actual file transfer process, the user needs a generic USB flash drive, where the offline stored data files can be copied, and a proper USB cable to connect the flash drive. The proposed offline data transfer scenario is illustrated in the sequence diagram (see Figure 16) below. Processes/ actions inside the block titled "**3.**" are being executed automatically by several *Python* and Linux shell scripts (no human interaction is required within the block #**3**).

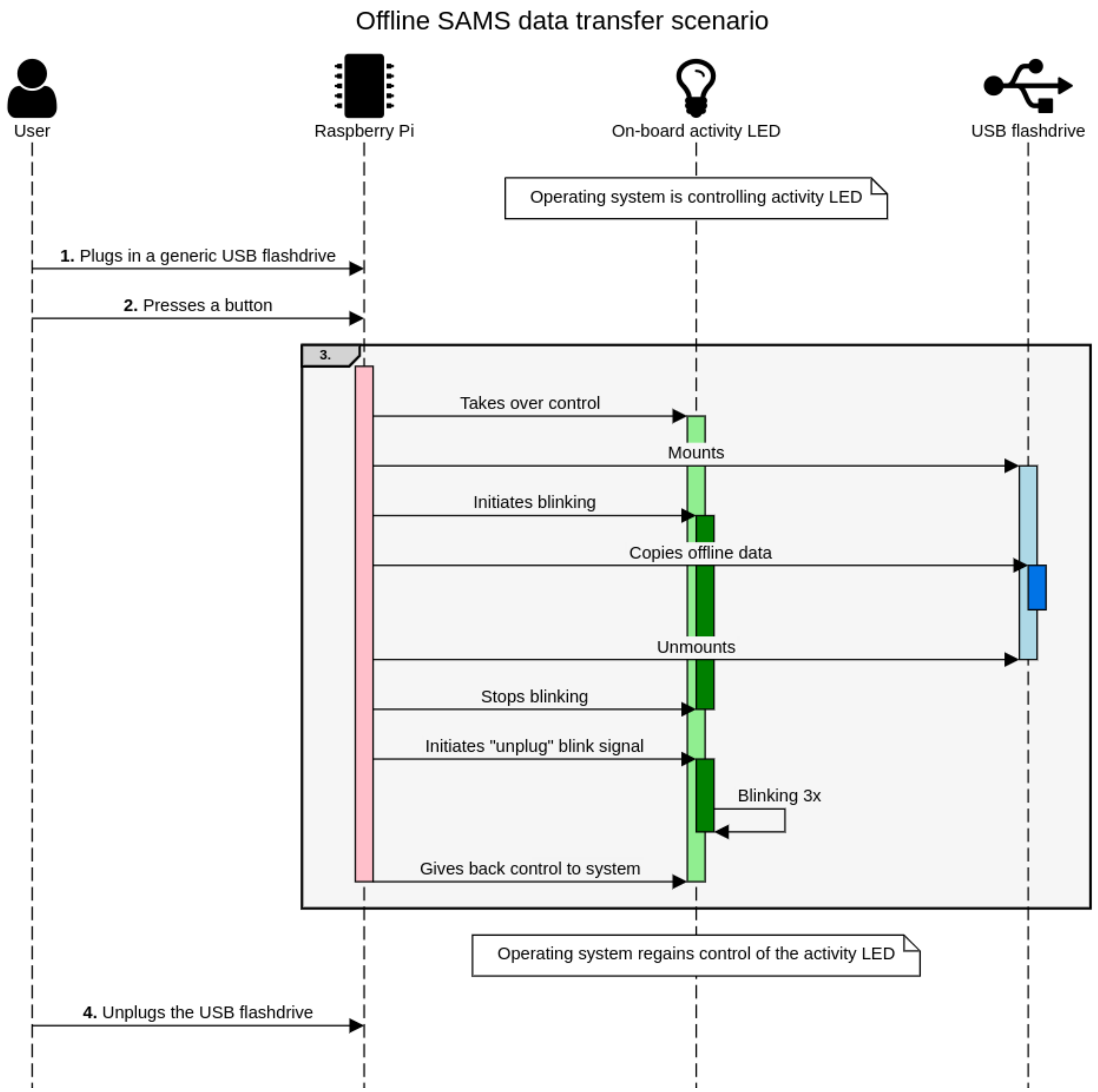

Offline SAMS data transfer scenario
User
Raspberry Pi
On-board activity LED
USB flashdrive
Operating system is controlling activity LED
1. Plugs in a generic USB flashdrive
2. Presses a button
3.
Takes over control
Mounts
Initiates blinking
Copies offline data
Unmounts
Stops blinking
Initiates "unplug" blink signal
Blinking 3x
Gives back control to system
Operating system regains control of the activity LED
4. Unplugs the USB flashdrive

Figure 16. Sequence diagram of the proposed offline SAMS data transfer scenario

The first step for the user is to plug in (**Step #1**) the USB flash drive to the Raspberry Pi Zero W. To initiate the file transfer process user should send a signal (preferably by pressing a push-button) to the Raspberry Pi Zero W (**Step #2**). A script running on the Raspberry Pi Zero W is continuously listening to a specific input pin (in an ideal situation, the mentioned push-button is attached to it) and monitors when the signal on the pin changes to the LOW state. When this signal is received, the script (**Steps #3**) takes over the control of the built-in activity LED that is internally connected to *GPIO47* of the Raspberry Pi's processor (*BCM2835*). The commands needed for the "take over" are as follows:

#To disable activity on act_led

```
echo none | sudo tee /sys/class/leds/led0/trigger
```

#To control (set the corresponding pin as output) the act_led within a Python script

```
act_led = 47

GPIO.setup(act_led, GPIO.OUT)
```

#To turn the act_led on

```
      GPIO.output(act_led, 0)
```

#To turn the act_led off

```
      GPIO.output(act_led, 1)
```

After the act_led is configured to be controlled, the USB flash drive is mounted by executing the following command:

```
      sudo mount $USB_DEV $USB_MOUNTPOINT -o uid=$USER,gid=$USER
```

When the script checks for a successful USB device's mount, the necessary files are being copied and the act_led starts to blink quickly, indicating the file transfer process is in progress. After the files have been copied successfully, the USB flash drive is unmounted and act_led blinks slowly 3 (three) times indicating that the flash drive can be safely removed. At the end the script gives back the control of the act_led to the operating system by executing:

```
      echo mmc0 | sudo tee /sys/class/leds/led0/trigger
```

It should be mentioned that the written script also checks for several possible errors, like if the flash drive is not plugged in and the push-button is being pressed (in which case the act_led only blinks three times).

4. Sustainability of the SAMS data warehouse

At this moment the SAMS DW is hosted on the local LLU server. After the project ends, it is planned to continue the partnership with Latvia University of Life Sciences and Technologies, Faculty of Information Technologies, to continue hosting the SAMS DW. Already connected hardware devices will still be able to send data to the DW. Furthermore, it will still be possible to link new devices to the Data Warehouse.

Also, detailed instruction on how to host the SAMS DW on different infrastructure will be developed, thus allowing any interested individual to host their own SAMS DW.

At this point several third parties were interested to test the SAMS DW by linking their developed bee colony monitoring systems to it.

The SAMS DW is built as an open-source platform and can be freely used after the project ends. Information below summarises the DW hosting process step by step.

DW solution has four main components:

- MongoDB database,
- DW Core,
- DW Web API,
- DW Front-end (single page application, SPA).

For an easier deployment and maintenance, it is recommended to use additional infrastructure components:

- Nginx used as a single point of access for external systems (serving static Front-end files and proxying requests to Web API);
- Docker used for deployment of the DW services and database.

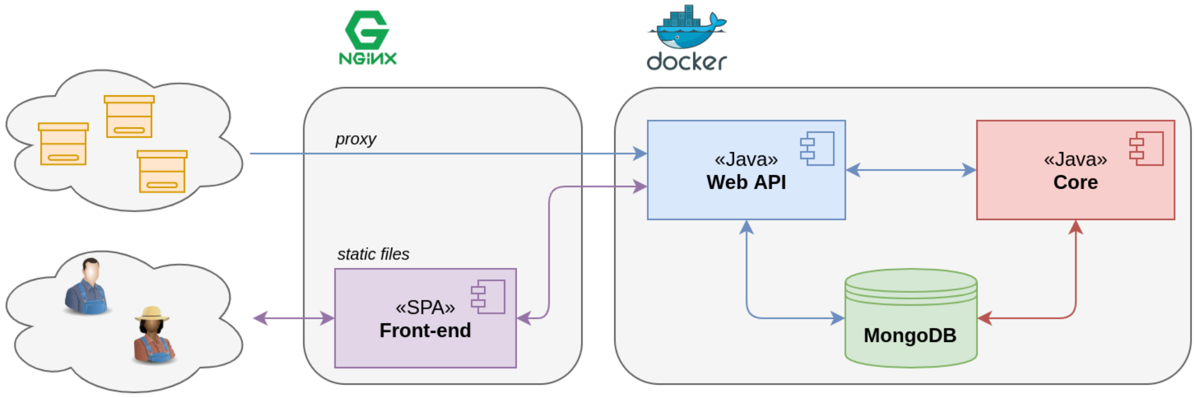


Figure 17. Architecture of the SAMS DW components

4.1 MongoDB

SAMS DW solution uses MongoDB database which is compatible with version 4. It has two separate databases for Web API and DW Core. There are no specific requirements for the MongoDB instance. In case of need it can be transparently replicated and/or shared across multiple hosts.

4.2 DW Core

DW Core provides the main data storage, analysis and reporting functionality. It can be built from the source code by running following command in the root of repository:

```
user@host:~/dwh-core$ ./gradlew core-bees:build
```

Compiled binary can be executed as follows:

```
user@host:~/dwh-core$  java  -jar  core-bees/build/libs/bees-dwh-core-{version}.jar
```

MongoDB instance is expected to be accessible on `localhost:27017` (can be changed via `spring.data.mongodb.uri` environment variable).

4.3 DW Web API

DW Web API provides restful API for all user and hardware related activities, like registering SAMS devices and source mappings, managing workspaces, receiving data from SAMS devices, providing data for the user interface including reports.

It can be built from the source code by running following command in the root of repository:

```
user@host:~/bees-dwh$ ./gradlew web-api:build
```

Compiled binary can be executed as follows:

```
user@host:~/bees-dwh$ java -jar web-api/build/libs/bees-dwh-web-api-{version}.jar
```

MongoDB instance is expected to be accessible on `localhost:27017` (can be changed via `spring.data.mongodb.uri` environment variable).

By default, it is expected that DW Core is running on the same host port `8081`. This this can be changed by the altering `dwh.address` environment variable.

4.4 DW Front-end

DW Front-end is a single page web application providing basic functionalities for the users. It can be built from the source code by running following commands in the root of repository:

```
user@host:~/bees-dwh$ ./gradlew front-end:build -PdownloadNode
```

In case that there is an available local Node.js installation `–PdownloadNode` parameter can be omitted.

Compiled web application is zipped and available in following directory:

```
user@host:$ ~/bees-dwh/front-end/dist/bees-dwh-front-end-{version}.tar.gz
```

Content of the archive can be directly served by any static resource host. By default, DW Front-end expects that the web API is available on the same host with URLs starting with /api, for example:

- https://sams.science.itf.llu.lv/ - Front-end Web application
- https://sams.science.itf.llu.lv/api/ - Web API service

This can be achieved by routing appropriate requests to web API (see example below).

4.5 Docker environment

For easier maintenance of all interrelated services it is recommended to use the orchestration software. Within the SAMS project Docker platform is used for such purposes, which was selected based on previous knowledge and expertise.

Docker is a set of platforms as a service (PaaS) products that uses OS-level virtualization to deliver software in packages called containers. Containers allow a developer to package up an application with all of the required parts, such as libraries and other dependencies, and ship it all out as one package.

In order to deploy all related services on one docker they have to be built as Docker containers.

Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines.

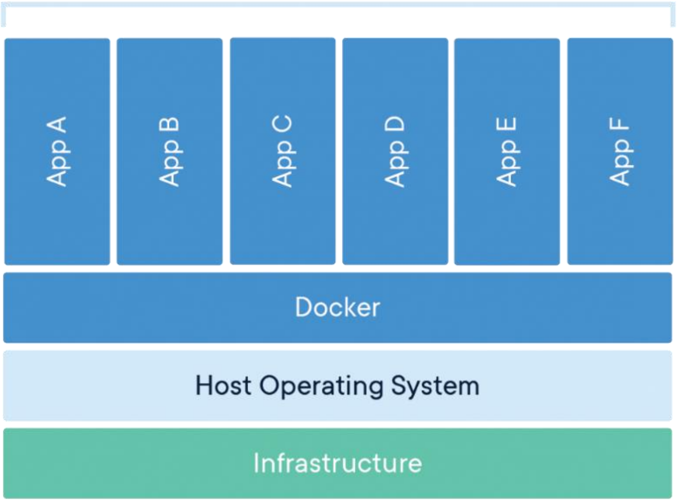


Figure 18. Concept of the containerized applications

There are a lot of options for how and where to run containers: AWS Elastic Container Service (AWS Fargate or a reserved instance with horizontal and vertical auto-scaling); a cloud instance with predefined a docker image in Azure or Google Cloud (with templates, instance groups, and auto-scaling); on your own server with a docker.

A local docker installation is needed for this purpose. The DW Core and Web API have appropriate commands:

```
user@host:~/dwh-core$ ./gradlew core-bees:dockerImage
user@host:~/bees-dwh$ ./gradlew web-api:dockerImage
```

This creates docker images on local environment. In case services are planned to be deployed on remote environments, it is recommended to use private docker image registry and push images to it.

```
user@host:~/dwh-core$ ./gradlew core-bees:dockerPush
user@host:~/bees-dwh$ ./gradlew web-api:dockerPush
```

For more information on how to run private registry refer to official documentation: https://docs.docker.com/registry/

The built images can be started as containers manually via `docker run` commands, or using `docker-compose` utility and provided configuration as follows:

```
user@host:~/bees-dwh$ docker-compose -f deploy/docker-compose.yml up
```

Configuration includes persistent volume for the data storage, private network for cross-service communication, deployment specific environment variables, etc.

4.6 Nginx

Nginx web server is used as a primary entry point for requests from the external internet. It is used both, for serving static DW front-end files as well as for routing requests to the DW web API container. Technically Nginx also could be deployed as a docker container, but within the SAMS project it is hosted directly on the server OS.

An example of Nginx configuration is available in file:

```
user@host:$ ~/bees-dwh/deploy/nginx.conf
```

Refer to official documentation for Nginx installation and configuration guide: https://nginx.org/en/docs/

4.7 Custom deployments

SAMS DW solution is created by considering current functional and hosting requirements. In case there is a need to use SAMS DW as a part of a custom (branded) solution and/ or host it on a different platform, several changes should be considered.

Credential handling

The SAMS DW solution uses Auth0 service for handling user and device credentials and relies on authentication routines provided by Auth0. For custom solutions dedicated Auth0 tenants should be registered by maintainers, or custom authentication solutions must be implemented both on DW front-end and DW web API components.

Infrastructure

Depending on the target host platform it might be needed to change hostnames and ports for particularly SAMS DW services. The current SAMS DW is hosted on LLU infrastructure and is maintained by the LLU staff.

Project website: www.sams-project.eu

Project Coordinator contact:

Stefanie Schädlich
Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH
Wielinger Straße 52
82340 Feldafing, Germany
stefanie.schaedlich@giz.de

DISCLAIMER

Neither GIZ nor any other consortium member nor the authors will accept any liability at any time for any kind of damage or loss that might occur to anybody from referring to this document. In addition, neither the European Commission nor the Agencies (or any person acting on their behalf) can be held responsible for the use made of the information provided in this document.