ZAFENA

20200221 13:31
ZAFENA
Acceptera & skicka in
DATALOGIC
QuickScan
M
TD-3128
SHOULDER
LEFT ARM

Workstation to cloud connectivity

Zafena POC-Workstation connects instruments, meters, sensors, label-printers and network-printers to the cloud system.

Once the Workstation is connected to the cloud system, all meters / sensors / instruments and printers connected to the Zafena Workstation now become, without further action, connected to the cloud system.

Analytical results, expressed in conformance with the HL7 FHIR standard, are forwarded by the Workstation to the cloud system.

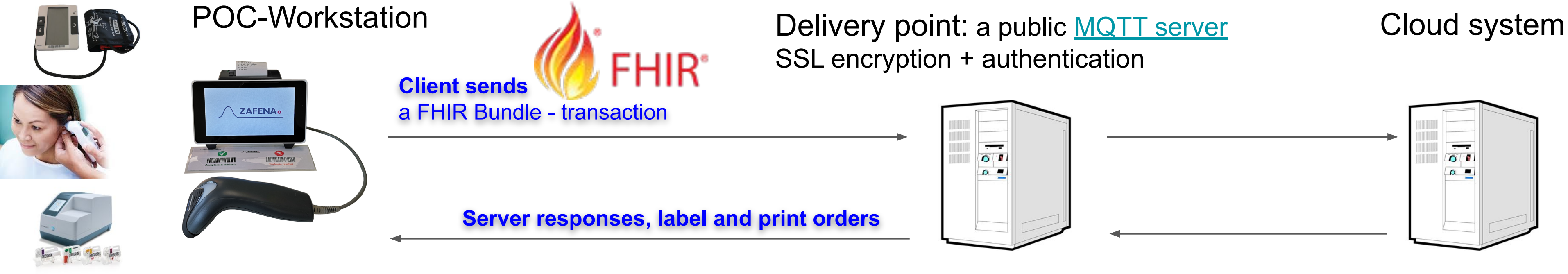
The analytes are categorized by international LOINC codes, or by NPU codes, to identify the type of analytical test and method.

Zafena recommends that the cloud operator uses a standardised delivery point.

Recommendation: The cloud system operates a public accessible delivery point, typically a MQTT server using SSL, all in conformance to standards.

- Zafena recommends the cloud operator to use the Continua Design Guidelines. The Continua design guideline standard define how to use FHIR in combination with HTTPs SSL (for **provisioning**) and MQTT SSL (for delivery of **FHIR results**) https://www.pchalliance.org/continua-design-guidelines

- The use of the MQTT server allows use of additional messages to cover use-cases outside of hospital standards: The additional messages allow **label printouts (.json) and print-orders (.pdf)**



POC-Workstation

Client sends a FHIR Bundle - transaction

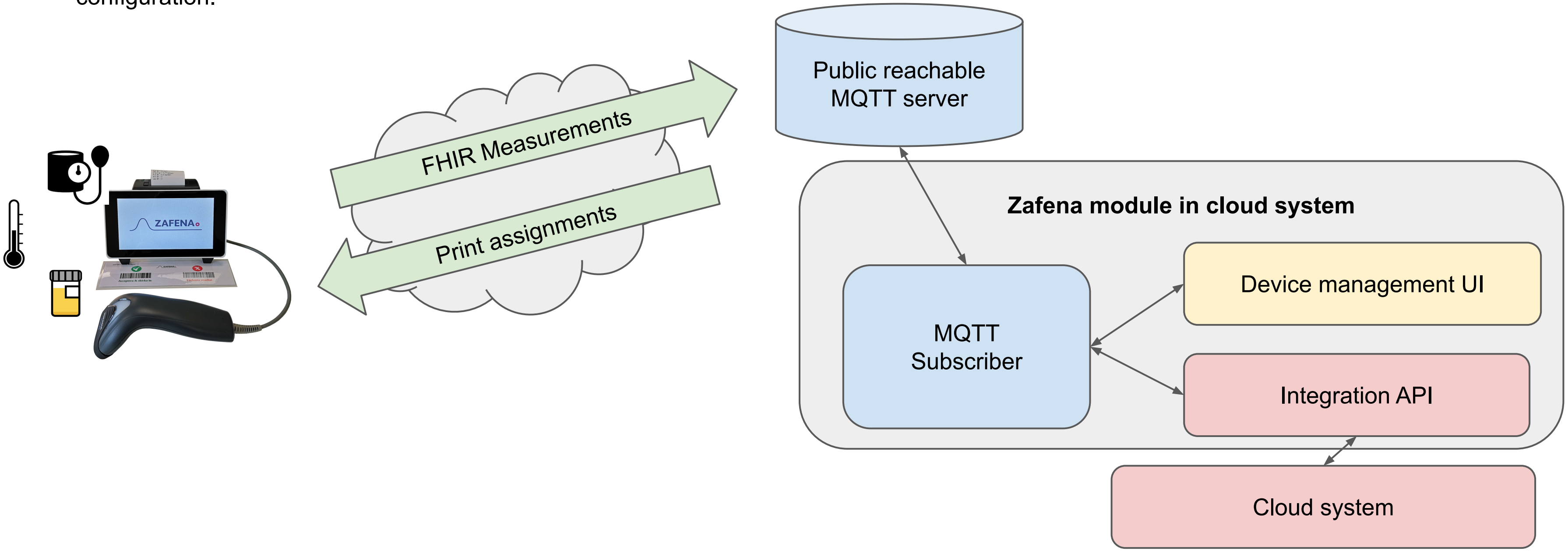
Delivery point: a public MQTT server
SSL encryption + authentication

Cloud system

Server responses, label and print orders

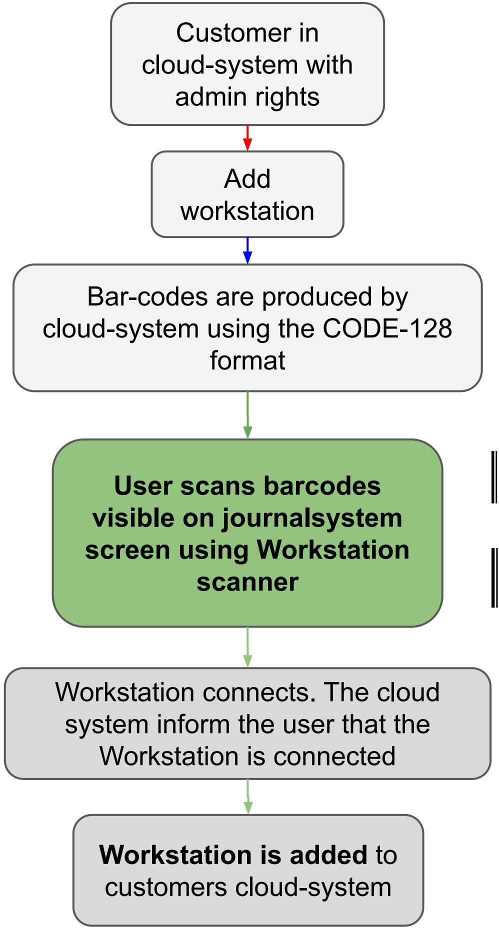
FHIR messages and response messages can be transmitted using MQTT
Zafena uses MQTT to support both FHIR messages and Cloud print orders.

- Measurements is sent as FHIR over MQTT
- Print assignments is sent as PDF over MQTT
- Label prints is sent at JSON over MQTT
- Device management UI keeps track of subscribed Zafena devices. It can be used for debugging, support and configuration.



Provisioning

Adding a new workstation to the customers clinic



Customer in cloud-system with admin rights

Add workstation

Bar-codes are produced by cloud-system using the CODE-128 format

User scans barcodes visible on journalsystem screen using Workstation scanner

Workstation connects. The cloud system inform the user that the Workstation is connected

Workstation is added to customers cloud-system

CFG-cloudLogin=abcderf

CFG-cloudPass=12345678

Since each Zafena workstation can be configured to suit individual needs, the workstations should be registered under the customer who has them in the cloud-system.

This can be done by adding a function to register installed workstations in the cloud-system.
Registration should be possible by customer (with admin rights) in their "own" system in the cloud-system.

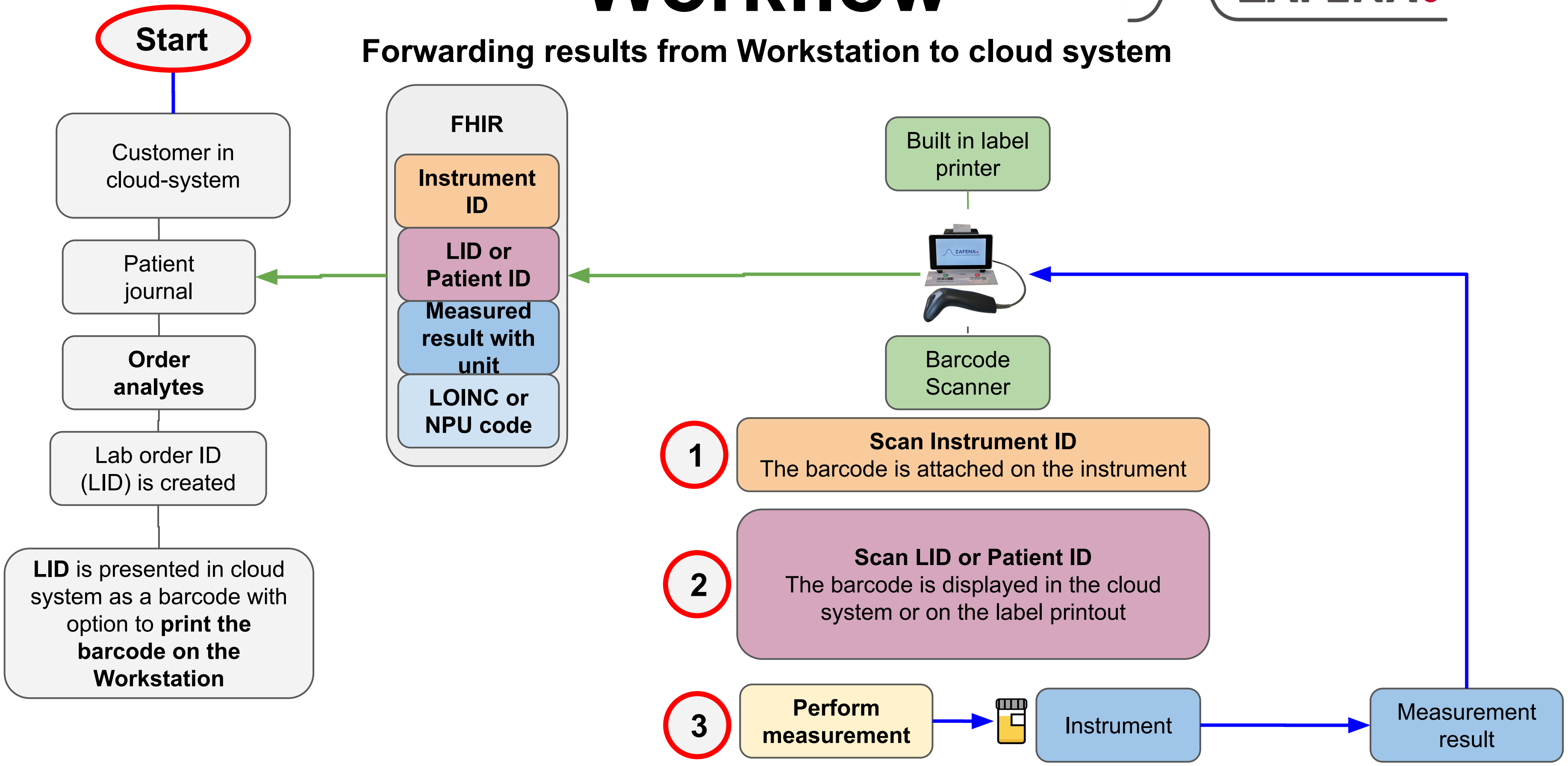
As a customer, you should have a webpage where you can add more workstations, but also see the ones you already installed, and be able to verify network access (ping). There should also be a test print function to verify that function.

The cloud server and encryption certificates are already configured into the Workstation when the Workstation is delivered to the customer. Therefore **the customer only need to scan the login and password** that are used **to connect the Workstation to the customers clinic** in the cloud-system.

Workflow

ZAFENA

Forwarding results from Workstation to cloud system



Start

Customer in cloud-system

Patient journal

Order analytes

Lab order ID (LID) is created

LID is presented in cloud system as a barcode with option to **print the barcode on the Workstation**

FHIR

Instrument ID

LID or Patient ID

Measured result with unit

LOINC or NPU code

Built in label printer

Barcode Scanner

1 **Scan Instrument ID**
The barcode is attached on the instrument

2 **Scan LID or Patient ID**
The barcode is displayed in the cloud system or on the label printout

3 **Perform measurement**

Instrument

Measurement result

Workflow - order to analyte

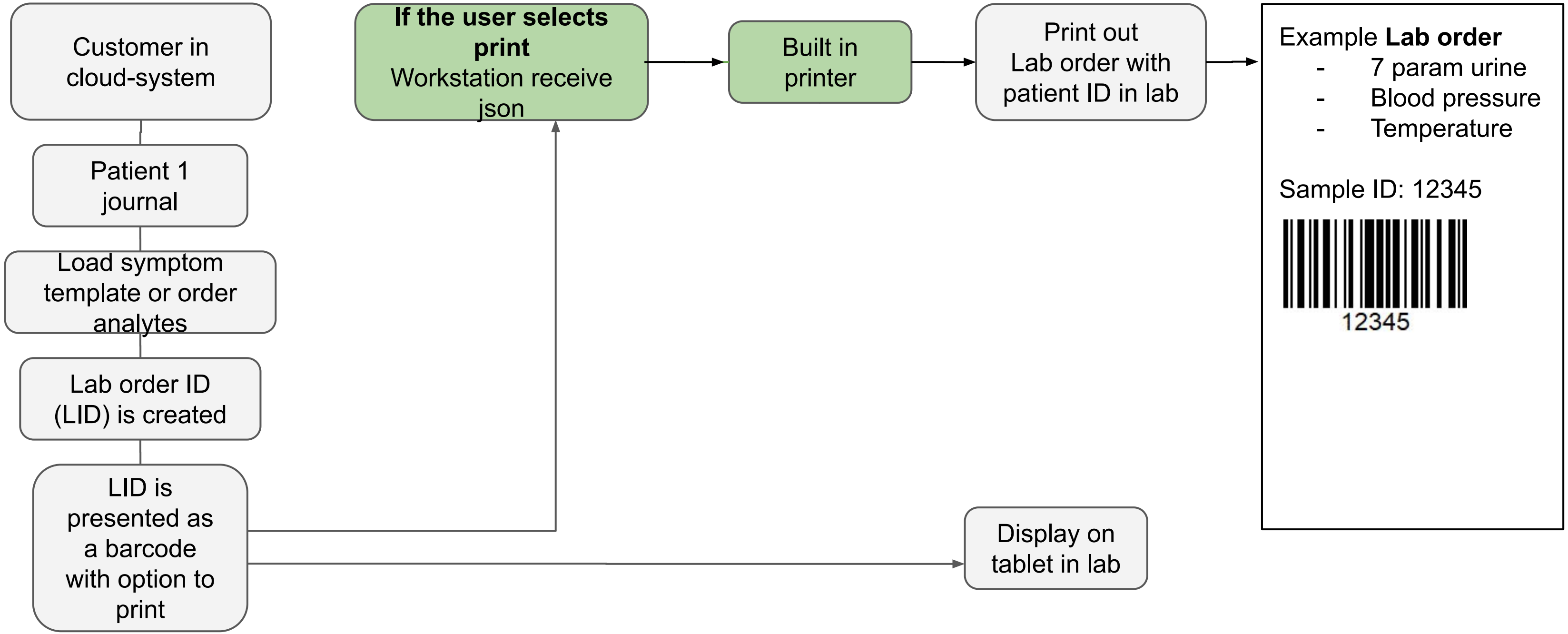
Use the barcode scanner connected to the Zafena workstation to scan the instrument-tag on preferred instrument.
The software for that specific instrument is shown on the workstation ready to receive analytical results.

The software will demand the operator to link the test to a Lab-order-ID (LID).
The ID is added by scanning the LID print-out, the content of the LID print-out is from the Cloud system.

When all demanded extra information is added, such as operator ID, the operator will perform the analysis and see that the result from the instrument will show up on the workstation.

An accept “OK” from the operator will send the measurement Result with the LID to the cloud-system receiver.

LAB ID creation



Customer in cloud-system

Patient 1 journal

Load symptom template or order analytes

Lab order ID (LID) is created

LID is presented as a barcode with option to print

If the user selects print
Workstation receive json

Built in printer

Print out
Lab order with patient ID in lab

Example **Lab order**

- 7 param urine
- Blood pressure
- Temperature

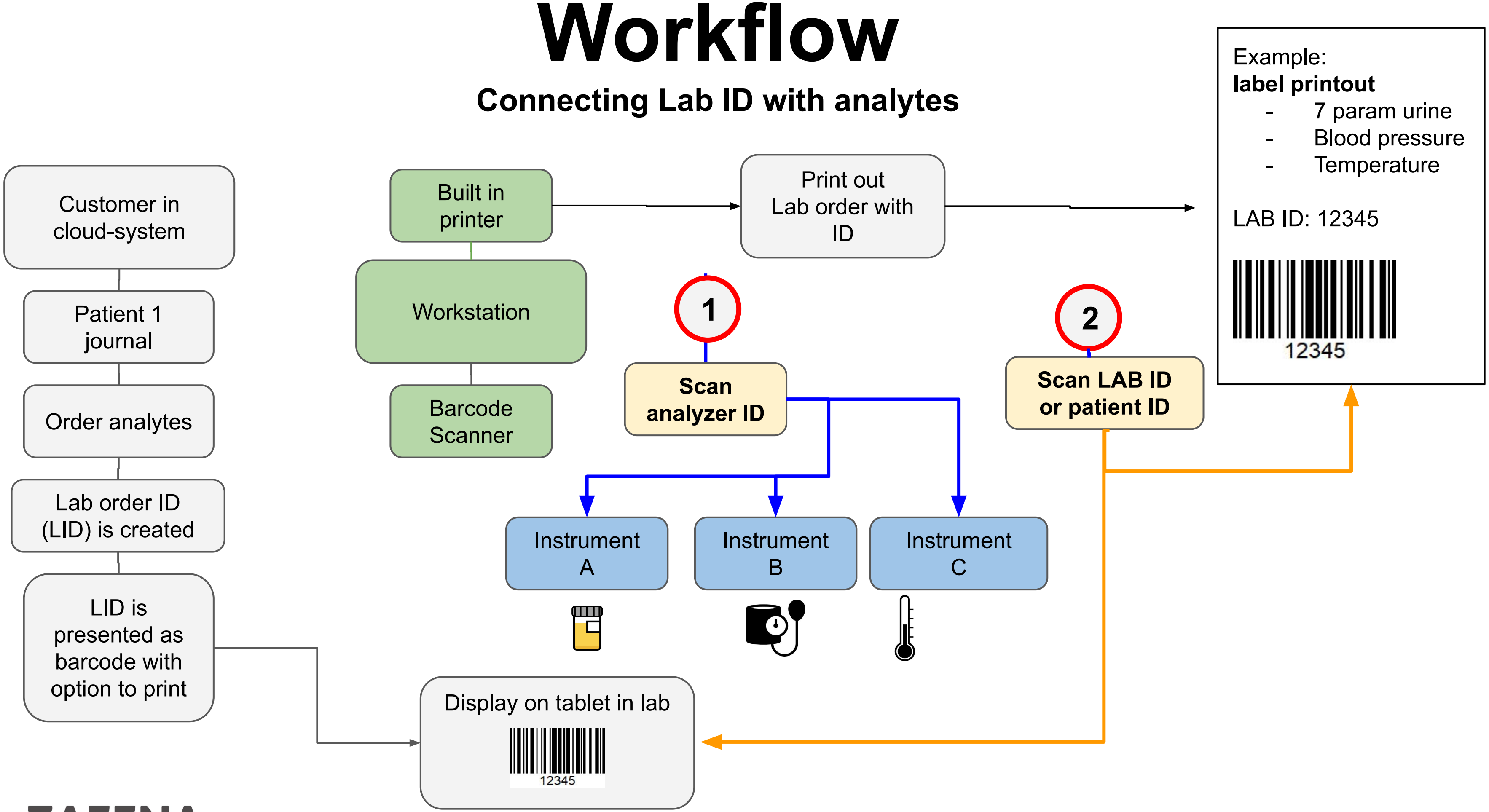
Sample ID: 12345

12345

Display on tablet in lab

Workflow

Connecting Lab ID with analytes



Customer in cloud-system

Patient 1 journal

Order analytes

Lab order ID (LID) is created

LID is presented as barcode with option to print

Built in printer

Workstation

Barcode Scanner

Print out Lab order with ID

1

Scan analyzer ID

2

Scan LAB ID or patient ID

Instrument A

Instrument B

Instrument C

Display on tablet in lab

12345

Example:
label printout

- 7 param urine
- Blood pressure
- Temperature

LAB ID: 12345

12345

ZAFENA

FHIR message in JSON example: Ear temperature

```
{"resourceType":"Bundle","entry":[{
"resource":{"resourceType":"Observation",
"Contained":[
{"resourceType":"Practitioner","id":"1","identifier":[{"value":"XRS1"}]},
{"resourceType":"Specimen","id":"2","identifier":[{"value":"84256"}],"type":{"coding":[{"system":"http://hl7.org/fhir/v2/0487","code":"EAR"}]}},
{"resourceType":"Device","id":"3", “serialNumber”:”12345678”}
],
"Identifier":[{"value":"84256"}],
"Status":"final",
"code":{"coding":[{"system":"http://loinc.org","code":"76011-6","display":"Ear temperature"}]},
"issued":"2020-02-26T14:16:00.000+01:00",
"performer":[{"reference":"#1"}],
"valueQuantity":{"value":36.5,"unit":"Cel"},
"specimen":{"reference":"#2"},
"device":{"reference":"#3"}}
}]}
```

Bundle = the complete FHIR message

Observation = the result

Each analyte gets assigned a LOINC code.
It is possible to use NPU codes if prefered.
The Identifier is the LID used by the cloud system to put the result in the right journal.

Contained = resources used by the Observation that are part of the same bundle.
The Observation #reference to the contained resource id’s.

https://www.hl7.org/fhir/bundle.html

https://www.hl7.org/fhir/Observation.html - When parsing FHIR Observation note that the actual value is 1 of 11 types. Zafena only use 2 types.

Zafena uses valueQuantity for all measurements with an Unit.

Zafena uses valueString for all measurements without an Unit.

https://www.hl7.org/fhir/datatypes.html#Quantity

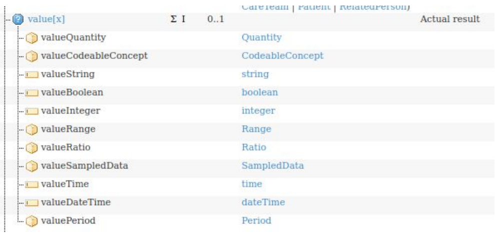
https://www.hl7.org/fhir/datatypes.html#string

https://www.hl7.org/fhir/references.html - Explains Contained resources

https://www.hl7.org/fhir/Specimen.html - May contain extra information about the sample, “EAR”

https://www.hl7.org/fhir/Practitioner.html - May contain extra information such as HSA ID

https://www.hl7.org/fhir/Device.html - May contain extra information such as device serial number



			CareTeam | Patient | RelatedPerson)	
value[x]	Σ I	0..1		Actual result
valueQuantity			Quantity	
valueCodeableConcept			CodeableConcept	
valueString			string	
valueBoolean			boolean	
valueInteger			integer	
valueRange			Range	
valueRatio			Ratio	
valueSampledData			SampledData	
valueTime			time	
valueDateTime			dateTime	
valuePeriod			Period	

FHIR message in JSON example: CRP

```
{"entry":[{"resource":{
"resourceType":"Observation",
"contained":[{"resourceType":"Practitioner","id":"1","identifier":[{"value":"XRS1"}]},
{"resourceType":"Specimen","id":"2","identifier":[{"value":"121212121"}],"type":{"coding":[{"system":"http://hl7.org/fhir/v2/0487","code":"BLDC"}]}},
{"resourceType":"Device","id":"3"}],
"identifier":[{"value":"88888"}],
"Status":"final",
"code":{"coding":[{"system":"http://www.npu-terminology.org","code":"NPU19748","display":"CRP"}]},
"issued":"2020-03-11T10:24:00.000+01:00",
"performer":[{"reference":"#1"}],
"valueQuantity":{"value":5,"comparator":"<","unit":"mg/L"},
"specimen":{"reference":"#2"},
"device":{"reference":"#3"}
}}],"resourceType":"Bundle"}
```

Bundle = the complete FHIR message

Observation = the result

Each analyte gets assigned a LOINC code.
It is possible to use NPU codes if prefered.
The Identifier is the LID used by the cloud system to put the result in the right journal.

Contained = resources used by the Observation that are part of the same bundle.
The Observation #references to the contained resource id's.

https://www.hl7.org/fhir/bundle.html

https://www.hl7.org/fhir/Observation.html - When parsing FHIR Observation note that the actual value is 1 of 11 types. Zafena [illegible] 2 t

Zafena uses valueQuantity for all measurements with an Unit.

Zafena uses valueString for all measurements without an Unit.

https://www.hl7.org/fhir/datatypes.html#Quantity

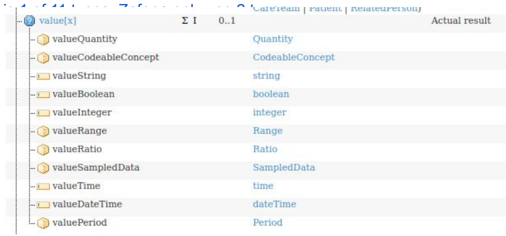
https://www.hl7.org/fhir/datatypes.html#string

https://www.hl7.org/fhir/references.html - Explains Contained resources

https://www.hl7.org/fhir/Specimen.html - May contain extra information about the sample, "BLDC"

https://www.hl7.org/fhir/Practitioner.html - May contain extra information such as HSA ID

https://www.hl7.org/fhir/Device.html - May contain extra information such as device serial number



			CareTeam | Patient | RelatedPerson)	
value[x]	Σ I	0..1		Actual result
valueQuantity			Quantity	
valueCodeableConcept			CodeableConcept	
valueString			string	
valueBoolean			boolean	
valueInteger			integer	
valueRange			Range	
valueRatio			Ratio	
valueSampledData			SampledData	
valueTime			time	
valueDateTime			dateTime	
valuePeriod			Period	

FHIR message in JSON example: Bloodpressure

```
{"resourceType":"Bundle","entry":[

{"resource":{"resourceType":"Observation",
"contained":[{"resourceType":"Practitioner","id":"1","identifier":[{"value":"FRWE"}]},
{"resourceType":"Specimen","id":"2","identifier":[{"value":"7203060699"}],"type":{"coding":[{"system":"http://hl7.org/fhir/v2/0487","code":"BDY"}]}},
{"resourceType":"Device","id":"3"}],
"identifier":[{"value":"7203060699"}],
"Status":"final",
"code":{"coding":[{"system":"http://loinc.org","code":"8480-6","display":"BP sys"}]},
"issued":"2020-03-10T13:13:00.000+01:00",
"performer":[{"reference":"#1"}],
"valueQuantity":{"value":160,"unit":"mmHg"},
"specimen":{"reference":"#2"},"device":{"reference":"#3"}}},

{"resource":{"resourceType":"Observation","contained":[{"resourceType":"Practitioner","id":"1","identifier":[{"value":"FRWE"}]},
{"resourceType":"Specimen","id":"2","identifier":[{"value":"7203060699"}],"type":{"coding":[{"system":"http://hl7.org/fhir/v2/0487","code":"BDY"}]}},
{"resourceType":"Device","id":"3"}],
"identifier":[{"value":"7203060699"}],
"Status":"final",
"code":{"coding":[{"system":"http://loinc.org","code":"8462-4","display":"BP dia"}]},
"issued":"2020-03-10T13:13:00.000+01:00","performer":[{"reference":"#1"}],
"valueQuantity":{"value":126,"unit":"mmHg"},
"specimen":{"reference":"#2"},"device":{"reference":"#3"}}},

{"resource":{"resourceType":"Observation","contained":[{"resourceType":"Practitioner","id":"1","identifier":[{"value":"FRWE"}]},{"resourceType":"Specimen","id":"2","identifier":[{"value":"7203060699"}],"type":{"coding":[{"system":"http://hl7.org/fhir/v2/04
87","code":"BDY"}]}},{"resourceType":"Device","id":"3"}],"identifier":[{"value":"7203060699"}],"status":"final","code":{"coding":[{"system":"http://loinc.org","code":"8867-4","display":"{beats}/min"}]},"issued":"2020-03-10T13:13:00.000+01:00","performer"
:[{"reference":"#1"}],"valueString":"97","specimen":{"reference":"#2"},"device":{"reference":"#3"}}}
]}
```

Bundle = the complete FHIR message
One Bundle may contain more than one Observation!

Observation = one result

When more than one observation are using the same identifier then they are all produced at the same measurement order.

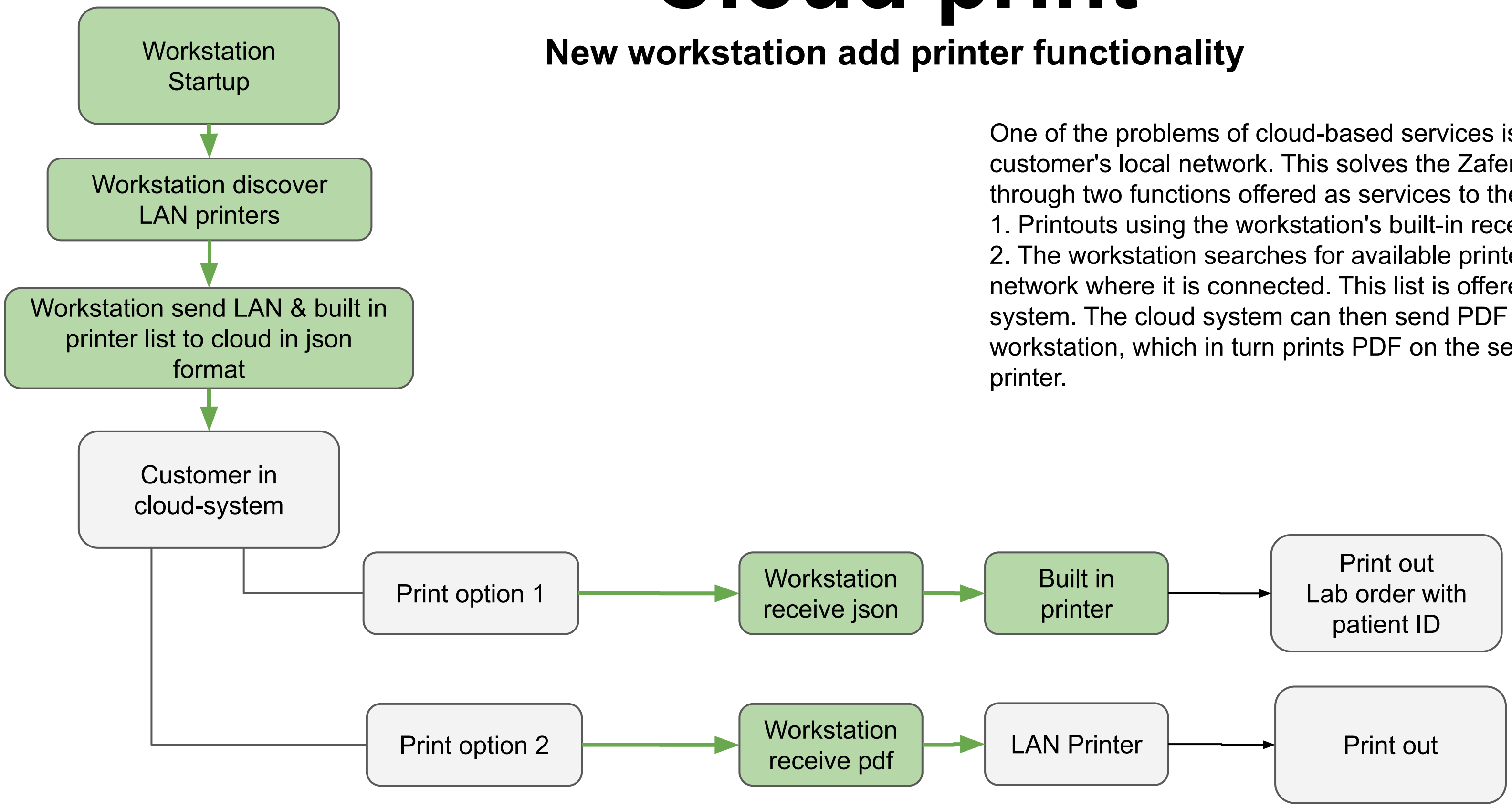
The third observation in this example is transferred using valueString. {beats}/min 97

Contained = resources used by the Observation that are part of the same bundle.
The Observation #references to the contained resource id's.

Cloud print

New workstation add printer functionality

One of the problems of cloud-based services is printing on the customer's local network. This solves the Zafena workstation through two functions offered as services to the cloud system.
1. Printouts using the workstation's built-in receipt printer.
2. The workstation searches for available printers on the same network where it is connected. This list is offered to the cloud system. The cloud system can then send PDF to the workstation, which in turn prints PDF on the selected network printer.



Workflow - results to cloud

When the operator “OK” the test, it will summon all data and send it through the FHIR protocol to the central cloud system. The data package will include:

LOINC code (analyte and method)
Analyte result (with unit)
Patient-ID (LID)
Instrument-ID (for traceability)
Time & date (time of measurement)

There are many open source reference implementations available to help implementers. Here are a list of the more common implementations used by implementers:

Full blown open source implementations for FHIR, some of which use these reference implementations, are listed on the HL7 wiki .

http://wiki.hl7.org/index.php?title=Open_Source_FHIR_implementations

Java HAPI-FHIR : Object Models, Parsers, Client + Server Framework, FHIR Validator, & Utilities. The FHIR specification is built with the HAPI-FHIR Java code. Zafena uses this reference implementation to construct FHIR messages.

C# HL7.FHIR : Object models, Parsers/Serialisers, Utilities, and a Client. Source code on GitHub at http://github.com/ewoutkramer/fhir-net-api

Pascal FhirServer : Object models, Parsers/Serialisers, Validator, Utilities, Client, and the FHIR Reference server. Requires Delphi (unicode versions)

Javascript FHIR.js : Javascript Client and Utilities

Swift Swift-FHIR : Object Model, Client and Utilities

JSON The cloud system may use a simple JSON parser to read the FHIR messages, the FHIR messages will always have the same structure.

Cloud print - detailed MQTT example

0: The Workstation connects and starts to listen on the device_ABCD/acl topic for topics in the cloud MQTT system to use.
The device_ABCD is the username used by the Workstation. The password is used when the workstation is connecting to the MQTT server.

1: The Workstation inform the cloud-system what it is configured for, the Workstation may send a new configuration at any time.

- one network printer
- one built in receit/label printer
- supports FHIR intrument measurements
- support a ping network test

The configuration is sent by writing a JSON message to the MQTT device_ABCD/config topic.

```
{"units": [{ "id": "instrument", "type": "fhir" }, { "id": "192.168.168.191", "type": "printer" }, { "id": "Prov-id skrivare device_ABCD", "type": "labeler" }, { "id": "connection_test", "type": "ping" }]}
```

2: The cloud-system respond on the device_ABCD/acl with a JSON list of what topics the Workstation should use:

```
["core/224/device_ABCD/instrument/fhir","core/224/device_ABCD/192.168.168.191/printer","core/224/device_ABCD/Prov-id skrivare device_ABCD /labeler","core/224/device_ABCD/connection_test/ping","core/224/device_ABCD/connection_test/pong"]
```

3: The Workstation starts listening for JSON print orders on the MQTT labeler topic
core/224/device_ABCD/Prov-id skrivare device_ABCD /labeler
The Workstation starts listening for PDF print orders on the MQTT printer topic
core/224/device_ABCD/192.168.168.191/printer
The Workstation know where to send FHIR measurements
core/224/device_ABCD/instrument/fhir
The Workstation know where to listen and where to send ping/pong for cloud to workstation network test
Listen on core/224/device_ABCD/test_instrument/ping and return the ping message to core/224/device_ABCD/test_instrument/pong

4: The Cloud-system sends what to print on the built in printer by posting a JSON message to the labeler topic:

```
[{"text":"Datum: 2020-03-05"},{"text":"Prov ID: 84258"},{"text":"Patient: Kristin Krank"},{"barcode":"84258"}]
```

ZAFENA