H.811 Personal Health Devices Interface Design Guidelines

Version 2016

August 4, 2016

Table of Contents

Figures

Tables

0 Introduction

The Continua Design Guidelines (CDG) defines a framework of underlying standards and criteria that ensure the interoperability of devices and data used for personal connected health services. It also contains additional design guidelines for interoperability that further clarify or reduce the options in underlying standards or specifications, or that add a feature missing in an underlying standard or specification. This guidelines document focuses on the **PHD-IF** (Personal Health Devices Interface).

This guideline is one of the "H.810 Interoperability design guidelines for personal health systems" documents. See [H.810] for more details.

0.1 Organization

This guideline document is organized in the following manner.

- **Clauses 0-5: Introduction and terminology**
  - These clauses provide useful background information to help understand these guidelines.
- **Clause 6: Common X73 PHD-IF design guidelines**
  - This clause provides an overview of the common elements of the PHD-IF architecture with design guidelines that apply to all Personal Health Devices (PHDs) and Personal Health Gateways (PHGs) implementing the PHD-IF using an IEEE 11073 PHD device specialization (X73 Device).
- **Clause 7: NFC design guidelines**
  - This clause is an overview of the NFC architecture along with the design guidelines for PHDs and PHGs that use NFC and IEEE 11073 PHD (X73) to implement the PHD-IF.
- **Clause 8: Bluetooth BR/EDR design guidelines**
  - This clause is an overview of the Bluetooth BR/EDR (basic rate / enhanced data rate) architecture along with the design guidelines for PHDs and PHGs that use Bluetooth BR/EDR and X73 to implement the PHD-IF.
- **Clause 9: USB design guidelines**
  - This clause is an overview of the USB architecture along with design guidelines for PHDs and PHGs that use USB and IEEE 11073 PHD to implement the PHD-IF.
- **Clause 10: ZigBee design guidelines**
  - This clause is an overview of the ZigBee architecture with design guidelines for PHDs and PHGs that use ZigBee and X73 to implement the PHD-IF.
- **Clause 11: Bluetooth LE design guidelines**
  - This clause is an overview of the Bluetooth LE (low energy) architecture along with the design guidelines for PHDs and PHGs that use Bluetooth LE to implement the PHD-IF. This clause does not refer to IEEE 11073 PHD.

0.2 Guideline releases and versioning

Information on releases and versioning of these guidelines can be found in Clause 0.2 of [H.810]

0.3 What's New?

To see what is new in this release of the design guidelines refer to Clause 0.3 of H.810 - *Interoperability design guidelines for personal health systems* [H.810].

1 Scope

This guidelines document focuses on the **PHD-IF** (Personal Health Devices Interface) that consists of the following sub-interfaces:

- **X73 Interface (X73-IF) –** interface based on IEEE/ISO 11073-20601 and a supported transport technology. Supported transport technologies are:
  - **NFC**
  - **Bluetooth BR/EDR**
  - **USB**
  - **ZigBee**

- **Bluetooth LE Interface (BLE-IF)** – interface based on a Bluetooth LE as transport, technology, and one or more (application level) services and profiles defined by Bluetooth SIG (Special Interest Group).

These interfaces are defined in the Continua architecture as described in [H.810], Clause 6 as shown in Figure 1-1 below.

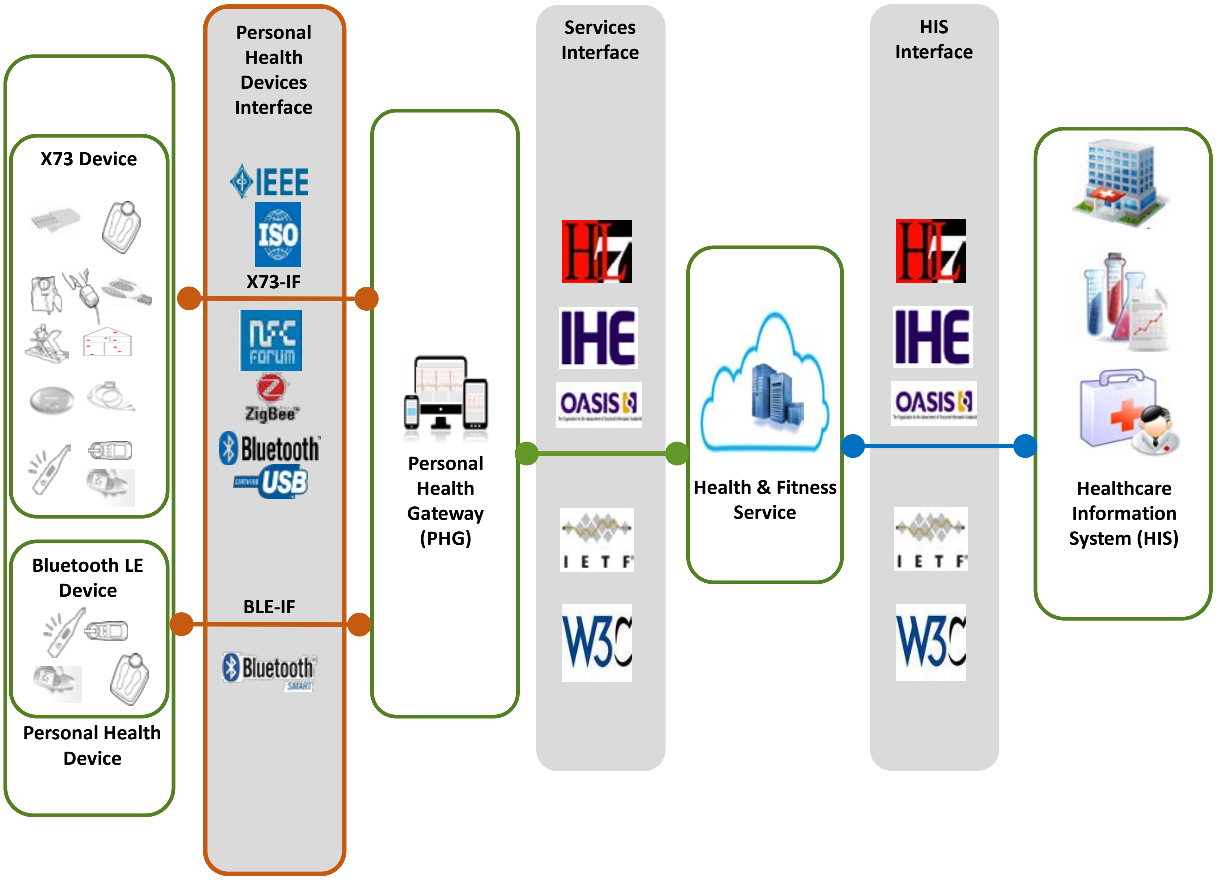

Figure 1-1 – Personal Health Devices Interfaces in the Continua architecture

These guidelines cover the following X73 Devices that can use one of the X73-IF supported transport technologies (ZigBee, NFC, USB and Bluetooth BR/EDR):

- activity hub,
- adherence monitor,
- basic 1-3 lead ECG sensor,

- blood pressure monitor,
- body composition analyzer,
- cardiovascular fitness,
- CO sensor,
- contact closure sensor,
- continuous glucose monitor,
- dosage sensor,
- enuresis sensor,
- fall sensor,
- gas sensor,
- glucose meter,
- heart-rate sensor,
- INR meter,
- insulin pump,
- motion sensor,
- peak flow meter,
- PERS sensor,
- property exit sensor,
- pulse oximeter,
- Sleep Apnea Breathing Therapy Equipment (SABTE),
- smoke sensor,
- step counter,
- strength fitness,
- switch sensor,
- temperature sensor,
- thermometer,
- usage sensor,
- water sensor,
- weighing-scales.

These guideline also cover a second group of Personal Health Device types that use the Bluetooth LE technology. This group consists of:

- blood pressure monitor,
- continuous glucose monitor,
- glucose meter,
- heart-rate sensor,
- pulse oximeter,
- thermometer,
- weight scale.

2 References

All referenced documents can be found in Clause 2 of H.810, *Interoperability design guidelines for personal health systems* [H.810].

3 Definitions

This guidelines document uses terms defined in [H.810].

4 Abbreviations and Acronyms

This guidelines document uses abbreviations and acronyms defined in [H.810].

5 Conventions

This guidelines document follows the conventions defined in [H.810].

6 Common X73 Personal Health Devices guidelines

NOTE: This clause (except for Clause 6.2.2.6) does not apply to Bluetooth LE Devices.

6.1 X73 interface architecture (informative)

6.1.1 Introduction

This clause lists the application layer design guidelines that are common to X73 PHD. This clause does not apply to the BLE-IF subclass of the PHD-IF (see Figure 6-2). See Clauses 7 to 10 for the guidelines per supported transport protocol.

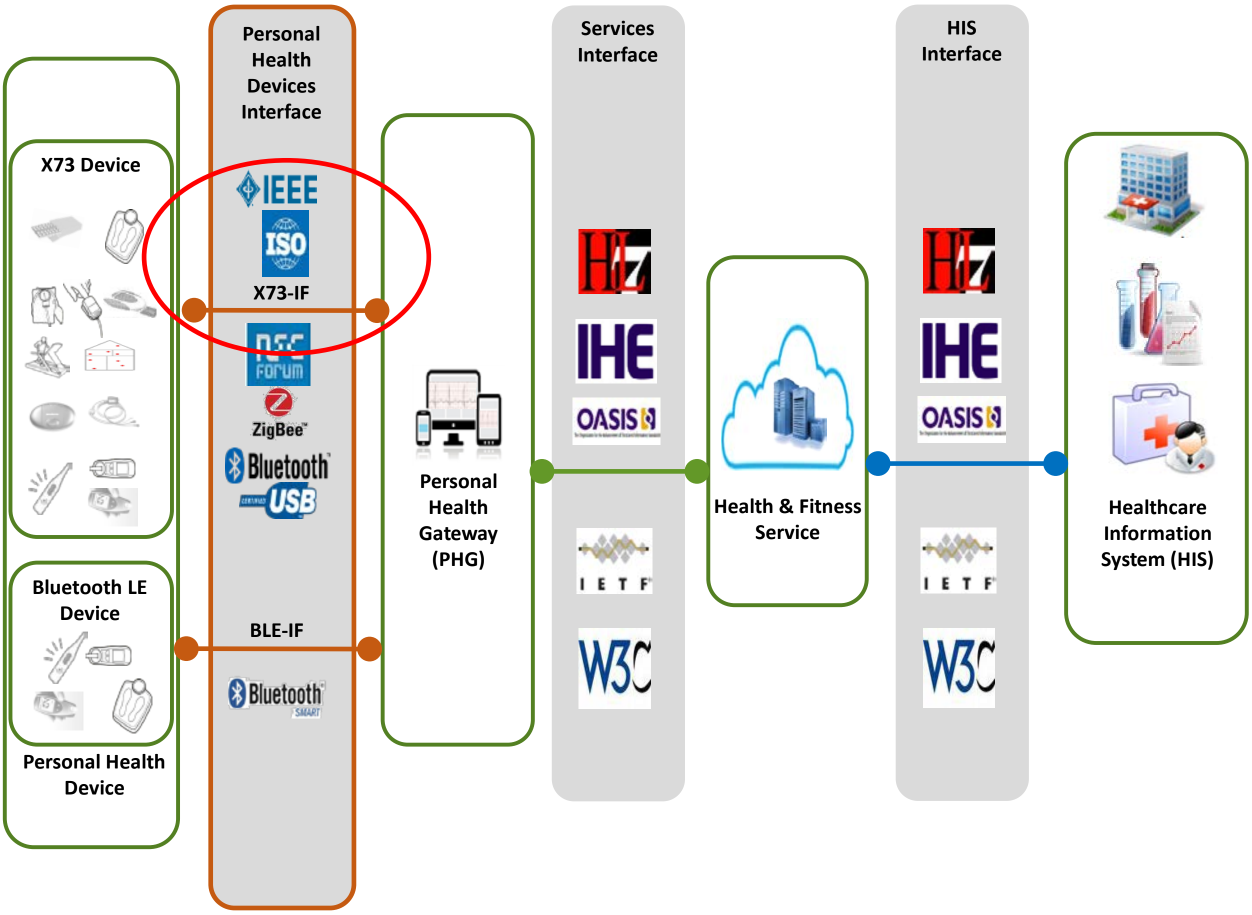

Figure 6-1 – X73-IF in the Continua E2E architecture

6.1.2 Overview

The X73-IF is composed of different layers. Appropriate standards are selected for the individual layers and establish interoperability in the personal health ecosystem. Figure 6-2 gives an overview of the protocol stack for the X73-IF.

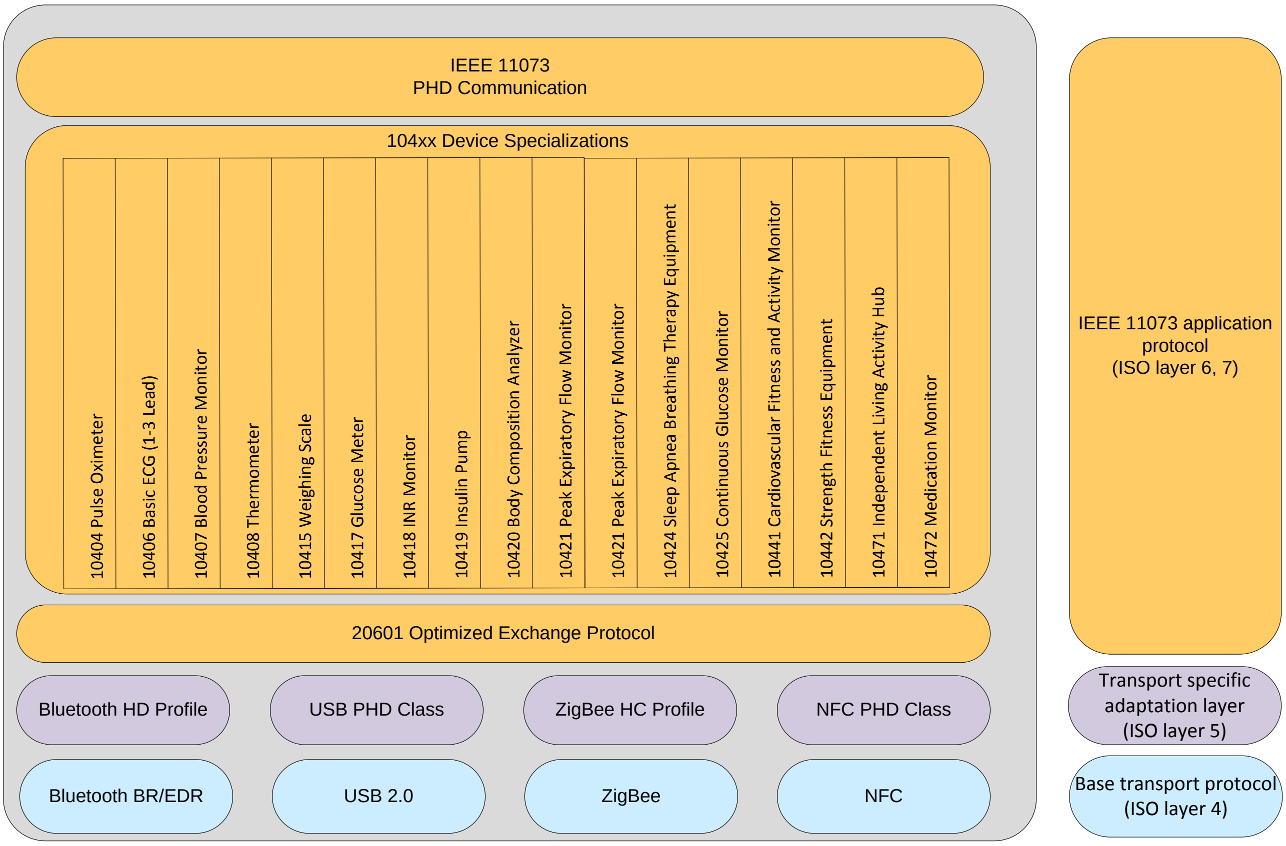

Figure 6-2 – X73-IF protocol stack

6.1.3 Common data/messaging layer and selected standards

Widely supported transport technologies and profiles have been selected for wireless and wired versions of the X73-IF. However, for the application level data/messaging there is considerable commonality. A common solution has been selected to serve as the data/messaging layer on top of the supported transport protocols.

The 11073-20601 Optimized Exchange Protocol described in [IEEE 11073-20601] has been selected as the basis of the application protocol for the X73-IF. This internationally harmonized standard provides an interoperable messaging protocol and has definitions and structures in place to convert from an abstract data format into a transmission format. Thus, a consistent data exchange layer is enabled for the X73-IF.

The IEEE 11073-20601 protocol (see [IEEE 11073-20601]) acts as a bridge between device-specific information defined in individual so-called device specializations and the underlying transports to provide a framework for optimized exchange of interoperable data units. The selected device specialization standards specify the data model and nomenclature terms to be used for individual devices. The device specializations are also illustrated in Figure 6-2.

6.1.4 Transport protocols and selected standards

The following wired and wireless solutions have been selected to serve as the CDG transport for the X73-IF:

– Bluetooth BR/EDR – Bluetooth Health Device Profile
– USB – USB Personal Healthcare Capability Class
– NFC – NFC Personal Health Device Communication Class

– ZigBee – ZigBee Health Care Profile

The selected protocols for the transport layer ensure interoperable set-up and tear-down of the communication channel for the transfer of control and data messages across all domains.

6.2 Common X73 layer guidelines

6.2.1 Applicable interfaces

This clause contains a general design guideline, in Table 6-1, that lists the CDG network interfaces for which the common data/messaging layer guidelines in Clauses 6.2.2 to 6.2.3 are applicable.

Table 6-1 – Applicable interfaces

Name	**Description**	**Comments**
11073-20601-Applicable-Interfaces	Continua X73-IF service and client components **shall** implement the guidelines in Table 6-2.	The referenced tables contain guidelines on the data/messaging layer, which are consistent for the listed interfaces. The BLE-IF uses a different data/messaging layer (see Clause 8.1.2).

6.2.2 Exchange protocol

6.2.2.1 X73 component - general

This clause contains general design guidelines, in Table 6-2, on the implementation of the [IEEE 11073-20601] specifications. All requirements in Clause 6.2.2 refer to these specifications.

Table 6-2 – X73 wired/wireless general requirements

Name	**Description**	**Comments**
11073-20601-Reqt	Continua X73-IF service and client components **shall** implement at least the version or versions of the [IEEE 11073-20601] specifications as defined in Table 6-3.	It depends on the supported device specializations and on the client or service role which version or versions of IEEE 11073-20601 shall at least be supported for a Certified Capability Class. Client and service components are allowed to implement multiple versions, with the minimum version(s) as specified in Table 6-3.

Name	Description	Comments
11073-20601-2010-BOT-Restriction	If the client component chooses to use protocol-version 1 in the association phase then the service component shall not use BO-time in the communication with this client.	[ISO/IEEE 11073-20601-2010] (Version 1) did not support base offset time, so all device specializations prior to CDG V2012 cannot support this attribute. This requirement guarantees backward compatibility and interoperability. The client (the manager) indicates it wants to use version 1 by setting (only) the version 1 bit of the protocol-version field in the Association Response message.
11073-20601-2010-BOT-Recommended	Continua X73-IF service components using [IEEE 11073-20601] protocol version 2 or higher **should** use BO-time when reporting time and time-stamps in events.	[IEEE 11073-20601] supports different flavors of time reporting. BO-time is the one that gives the best possibilities on handling local time changes, DST settings, and synchronization with UTC.
11073-20601A-Service-Proto-Version	Continua X73-IF service components using [IEEE 11073-20601] protocol version 1 in an association **shall** adhere to the corrections and clarifications from [IEEE 11073-20601A]	Components certifying to CDG 2012 (and later) are required to indicate supported protocol versions as per the standards. Since early Continua X73-IF service components require implementation of [ISO/IEEE 11073-20601-2010] with only the corrections and clarifications from [IEEE 11073-20601A], these interfaces will follow protocol version 1 (with corrections).
11073-20601A-Client-Proto-Version	Continua X73-IF client components using [IEEE 11073-20601] protocol version 1 in an association **shall** adhere to the corrections and clarifications from [IEEE 11073-20601A]	Responding to an AARQ with the version 1 bit of the protocol version set indicates that the Base Offset Time is not used. Similar to the Continua X73-IF service components, the Continua X73-IF client component shall nevertheless follow the remaining specifications of [IEEE 11073-20601A] even though the specification requires protocol version bit 2 to be set.
11073-20601A-Client-Other-Proto-Version	Continua X73-IF client components **may** accept other bit settings in the protocol version than the ones implied by Table 6-3, but would then be operating in a non-Continua certified association	This guideline allows Continua X73-IF client components to implement new technical extensions NOTE – This is outside the current Continua Certification Program.

Table 6-3 – Minimally supported base protocol version(s) for device specializations

IEEE 11073-104xx specification	device specialization(s)	20601 protocol version client component (*)	20601 protocol version service component (*)	specification supports BO-time
10404	pulse oximeter	v1	v1	no
10406	basic 1-3 lead ECG & heart-rate sensor	v2	v2	yes
10407	blood pressure monitor	v1	v1	no
10408	thermometer	v1	v1	no
10415	weighing scales	v1	v1	no
10417	glucose meter	v1	v1	no
10418	INR meter	v2	v2	yes
10419	insulin pump	v3	v3	yes
10420	body composition analyzer	v1	v1	no
10421	peak flow monitor	v1	v1	no
10424	sleep apnea breathing therapy equipment	v2	v2	yes
10425	continuous glucose monitor	v3	v3	yes
10441	cardiovascular fitness & step counter	v1, v2	v2	yes
10442	strength fitness	v1	v1	no
10471	activity hub, fall sensor, motion sensor, enuresis sensor, contact closure sensor, switch sensor, dosage sensor, water sensor, smoke sensor, property exit sensor, temperature sensor, usage sensor, PERS sensor, CO sensor & gas sensor	v1	v1	no
10472	adherence monitor	v1	v1	no

(*) The protocol versions “v1”, “v2” and “v3” as used in this table refer to the protocol version of the 11073-20601 protocol. The supported version(s) are indicated by bits in the “protocol –version” field included in the PHDAssociationInformation structure in the association request (AARQ) and association response (AARE) of the 11073-20601 protocol.

Table 6-4 – Correspondence between -20601 protocol versions and specifications

11073-20601 protocol version	Corresponding specification
v1	[ISO/IEEE 11073-20601-2010]
v2	[IEEE 11073-20601A]
v3	[IEEE 11073-20601-2014]

6.2.2.2 X73 component – communication capabilities

This clause contains guidelines for general communication capabilities of sensor components in Table 6-5, Table 6-6, Table 6-7 and Table 6-8.

Table 6-5 – Communication capabilities – general

Name	Description	Comments
11073-20601-Bidirectional	Continua X73-IF service and client components **shall** support bidirectional transmission (i.e., sending and receiving, of [IEEE 11073-20601] defined application layer messages)	
11073-Manager-Initiated-Communications	Continua X73-IF service components **shall not** support the MDS-Data-Request Action for the transfer of CDG data. This prohibits the service component from using manager initiated event reporting as a mechanism of measurement transfer	This guideline prohibits the use of manager-initiated event transmission. Use of this mechanism causes increased implementation and test effort that can be avoided through the use of a scanner. CDG data is defined as data from any object normatively defined in a device specialization
11073-DataReqMode-Alignment	Continua X73-IF service components **shall** ensure that the fields in the *Metric-Spec-Small* attribute of metric objects are aligned with what was declared in the DataReqModeCapab structure during Association	For example, if the *mss-acc-agent-initiated* bit is set in *Metric-Spec-Small*, then *data-req-init-agent-count* in *DataReqModeCapab* needs to be set to 1
11073-20601-FIFO-Store-and-Forward	Continua X73-IF service components that are designed to store and forward temporary measurements **shall** transmit data in a "First In First Out" sequence	This guideline applies to both temporarily stored measurement events and to measurement data stored in a PM-Store

Table 6-6 – Communication capabilities – event reporting

Name	Description	Comments
11073-20601-Config-Changes-Service	Continua X73-IF service components **shall** report configuration changes to future measurements only	In the context of these guidelines, configuration changes are changes to attributes that provide context for the measurement. The interpretation of the measurement depends on the values of these contextual attributes, or configuration values. An example of configuration change would be changing the unit code of the reported measurement (e.g. from pounds to kilograms)
11073-20601-Config-Changes-Client	Continua X73-IF client components that receive a report of a configuration change **shall** apply the change to future measurements only	A configuration update does not apply retroactively to data already received by the client component

Table 6-7 – Communication capabilities – scanner requirements

Name	Description	Comments
11073-20601-Scanner-Sole-Reporter	Continua X73-IF service components **shall** send changes to any particular attribute via a single scanner object (if enabled) or the MDS object, but never more than one object (of either the MDS or scanner type)	This guideline and the next one assigns responsibility to objects in the system for notifying the manager of changes and updates The scanner will report changes for attributes in the Scan-Handle-Attr-Val-Map
11073-20601-Unique-Scanner	Continua X73-IF client components **shall not** simultaneously turn on multiple scanners that embed the same measurement object provided by a single service component	

Table 6-8 – Communication capabilities – time setting

Name	Description	Comments
11073-20601-Set-Time	Continua X73-IF client components that receive a report containing the *Mds-Time-Info* attribute, with the mds-time-mgr-set-time bit set to 1, **shall** invoke the Set-Time action command within a TO_{config} time period in order to set the absolute time on the Continua X73-IF service component that has sent the report.	This guideline ensures the same client behaviour as for the case when the mds-time-mgr-set-time bit is received via a GET MDS response message (see [IEEE 11073-20601]).

Name	Description	Comments
11073-20601-DateAndTimeUpdate-PMSegmentTransfer-Server	Continua X73-IF service components that are in the middle of a PM-segment transfer **shall not** update the PM-Segment object *Date-and-Time–Adjustment* attribute regardless of any time changes that occur while the segment continues to be transferred"	This guideline ensures that the PM-segment includes measurements from the same, unbroken timeline. NOTE: This is somewhat less likely to occur at the USB/NFC/Bluetooth BR/EDR level since there is not programmatic control from another channel, but it could happen that the UI is still turned on during the transfer so this will cover this case
11073-20601-DateAndTimeUpdate-PMSegmentTransfer-Client	Continua X73-IF client components that receive a *Date-and-Time* update from a Continua X73-IF service component in the middle of a PM-segment transfer **shall** use the service component's time reference at the time the first segment entry is transmitted as the reference for the full segment regardless of any time changes that occur while the segment continues to be transferred	This guideline accounts for the fact that the service component's PM-segment contains measurements from the same, unbroken timeline.
11073-20601-DateAndTimeUpdate-PMSegment-LowResource-Service	Continua X73-IF service components with limited memory which implement PM-Store and do not implement Base-Offset-Time **may** maintain measurements across date or time adjustments within a single PM-segment. In this case, the user-facing time of the X73-IF service component at the time of the measurement **shall** be communicated as the measurement timestamp. See note 1 below.	In this case, such service components will not be capable of communicating date or time adjustments and cannot fulfill the requirement within [IEEE 11073-20601] Time Coordination section which states: “If an agent collects PM-store measurements and the Date-and-Time attribute is adjusted, the agent **shall** ensure that each PM-segment includes only measurements from the same, unbroken timeline”. This requirement is a fix that’s only valid for the current release of this document. In the next version it will be removed and all service components must properly handle time and date changes.

Note 1:

This requirement resolves the issue with some configurations of current IEEE device specializations that do use a PM Store with multiple segments and that do not include support for Base Offset time. Support of such a configuration would require an implementation to create new segments on each time or date change and to report on this in a single APDU as response to a GetSegmentInfo request from the manager. The memory needed to store the additional segments and the size of the response

APDU both grow significantly with each time or date change. This is seen as an unreasonable requirement on such implementations as they would run out of memory too quickly.
This affects configurations of the following device specializations that include a PM Store:

- Glucose Meter [IEEE-11073-10417-2011]
- Medication monitor [ISO/IEEE-11073-10472-2012]
- Pulse oximeter [ISO/IEEE-11073-10404-2010]

6.2.2.3 X73 component – device information

This clause contains design guidelines that describe how to map CDG required device information to [IEEE 11073-20601] defined attributes. These guidelines are covered in Table 6-9.

Table 6-9 – Device Information

Name	Description	Comments
11073-20601-Manufacturer	Continua X73-IF service components **shall** set the *manufacturer* field of the *System-Model* MDS object attribute to the device original manufacturer's name. If this capability is available, the *manufacturer* field **may** be overwritten to the customer-facing company's name by the customer-facing company	
11073-20601-Model	Continua X73-IF service components **shall** set the *model-number* field of the *System-Model* MDS object attribute to the device original manufacturer's model number. The *model-number* field **may** be overwritten to the customer-facing company's model by the customer-facing company	
11073-20601-OUI	The OUI part of the MDS *System-Id* attribute in a Continua X73-IF service component **shall** remain unchanged from the value set by the original manufacturer	This is a unique identifier, which is obtained by the IEEE registration authority and which is associated with a company. This attribute maps to the organizationally unique identifier (OUI) part (first 24 bits) of the EUI-64 attribute
11073-20601-DID	The 40 bit manufacturer defined identifier in the *System-Id* of the MDS object attribute of a Continua X73-IF service component **shall** remain unchanged from the value set by the original manufacturer	In combination with the System-Id attribute OUI part, this is a unique identifier associated with the device. It is required in order to facilitate data quality analysis. This attribute maps to the company-defined part (last 40 bits) of the EUI-64 attribute

Name	Description	Comments
11073-20601-DID-Bijective	There **shall not** be multiple different *System-Id* values that identify the same X73-IF service component	This guideline ensures that the System-Id value is an objective identifier of a device, i.e., in addition to every physical device having a globally unique identifier, each assigned identifier corresponds to a different physical device. As a consequence, a device cannot use multiple different System-Id values
11073-20601-Serial-Number	Continua X73-IF service components **shall** include a component to the *Production-Specification* MDS-object attribute with the *spec-type* field set to *serial-number* and the *prod-spec* field set to the serial number of the device	
11073-20601-FW-Revision	Continua X73-IF service components that provide a firmware identifier **shall** include a component to the *Production-Specification* MDS-object attribute with the *spec-type* field set to *fw-revision* and the *prod-spec* field set to the firmware identifier of the device	The firmware identifier is the version of the firmware deployed on the X73 Device. The firmware release deployed on an X73 Device is uniquely identified by the firmware identifier

6.2.2.4 X73 component – unsupported service component

The CDG provides the data and messaging information to enable interoperability between Personal Health Devices. However, there may be regulatory reasons that require some client components to be exclusive about the data they accept. Not all client components will need to be this exclusive. However, the CDG provides the data and the messages for client components that are exclusive to providing the user with a positive experience.

This clause contains design guidelines, in Table 6-10, that define the expected behaviour when a service-side certified capability is not available.

Table 6-10 – Unsupported service component

Name	Description	Comments
11073-Unsupported-Device-Rejection	If a Continua service component does not support at least one Continua Certified Capability Class supported by the client component and the client component only accepts Continua Certified Capability Classes, then the Continua X73-IF client components **shall** request to release the association with a Continua service component using a result field no-more-configurations	If the service component supports any Continua certified Capability Classes, it supports the corresponding Reg-Cert-Data-List MDS object attribute where the certified Capability Class will be listed. The client will need to query the MDS to retrieve this attribute. It is recommended that this query is done before the service component enters the operating state to avoid the unwanted transfer of data
11073-Unsupported-Device-Utilize-11073	Continua X73-IF service and client components that need to selectively accept or reject service or client component data for a specialization they support in order to comply with regulatory requirements **shall** utilize only [IEEE 11073-20601] data structures to make the decision to reject or accept data from a client or service component	It will be necessary to simulate "accepted" devices to fully test service and client components. Device manufacturers will need to document and provide 11073 data structures for "accepted" devices for use during interoperability testing. Note that this design guideline is not a testable design guideline. It is simply used to facilitate testing
11073-Unsupported-Device-UserNotification-Client	Continua X73-IF client components **shall** notify the user of failure of the connection and corresponding reason, if it has released or rejected the association according to requirement 11073-Unsupported-Device-Rejection	This requirement is related to the user interface of the client component. Notification can be done in various ways (e.g. by displaying a text message or by means of a blinking LED)
11073-Unsupported-Device-UserNotification-Service	Continua X73-IF service components **should** notify the user of failure of the connection and corresponding reason, if the client has released or rejected the association according to requirement 11073-Unsupported-Device-Rejection	This requirement is related to the user interface of the service/client component. Notification can be done in various ways (e.g. by displaying a text message or by means of a blinking LED)

Name	Description	Comments
11073-Unsupported-Device-UserNotification-String-Client	Continua X73-IF client components with appropriate UI capabilities **should** use the following text string to notify the user of the connection failure in accordance with guideline 11073-Unsupported-Device-UserNotification-Client: "Thank you for choosing Continua certified personal health products. The device you are connecting either has not been Continua certified or the data is not intended for use in this solution. Please see your user manual for more details."	This string **may** be localized by the manufacturer based on the product and target geography
11073-Unsupported-Device-UserNotification-String-Service	Continua X73-IF service components with appropriate UI capabilities **should** use the following text string to notify the user of any failure of the connection according to guideline 11073-Unsupported-Device-UserNotification-Service: "Thank you for choosing Continua certified personal health products. The device you are connecting either has not been Continua certified or the data is not intended for use in this solution. Please see your user manual for more details."	This string **may** be localized by the manufacturer based on the product and target geography
11073-Unsupported-Device-NotificationDocu	Continua X73-IF service and client components **shall** be shipped with a documentation of the notification mechanism with respect to requirements 11073-Unsupported-Device-UserNotification-Service and 11073-Unspported-Device-UserNotification-Client	

6.2.2.5 X73 component – quality of service

To send [IEEE 11073-20601] data and messages on logical channels based on QoS characteristics, the requirements in Table 6-11 are defined.

Table 6-11 – X73 QoS implementation

Name	Description	Comments
DataMessaging-BiDir-QoS	Continua X73-IF service and client components shall send all messages on the corresponding Continua QoS bins listed in Table 6-12	

Table 6-12 – Bidirectional transport layer: Message type/QoS bin mapping

Msg Grp	Message type description	APDU Type	QoS bin type
0	Association Request	Aarq	best.medium
	Association Response	Aare	best.medium
	Association Release Request	Rlrq	best.medium
	Association Release Response	Rlre	best.medium
	Association Abort	Abrt	best.medium
1	DATA(Invoke-**Un**confirmedEventReport (Unbuf-Scan-Report-*), ScanReportInfo*)	Prst	best.medium or good.medium
	DATA(Invoke-**Un**confirmedEventReport(Buf-Scan-Report-*), ScanReportInfo*)	Prst	best.medium or good.medium
	DATA(Invoke-**Un**confirmedEventReport (MDS-Dynamic-Data-Update-*), ScanReportInfo*)	Prst	best.medium or good.medium
2	DATA(Invoke-ConfirmedEventReport(MDS-Configuration-Event), ConfigReport)	Prst	best.medium
	DATA(Response-ConfirmedEventReport(MDS-Configuration-Event), ConfigReportRsp)	Prst	best.medium
	DATA(Invoke-ConfirmedEventReport(Segment-Data-Event), SegmentDataEvent)	Prst	best.medium
	DATA(Response-ConfirmedEventReport(Segment-Data-Event), SegmentDataResult)	Prst	best.medium
	DATA(Invoke-ConfirmedEventReport(Unbuf-Scan-Report-*), ScanReportInfo*)	Prst	best.medium
	DATA(Response-ConfirmedEventReport(Unbuf-Scan-Report-*))	Prst	best.medium
	DATA(Invoke-ConfirmedEventReport(Buf-Scan-Report-*), ScanReportInfo*)	Prst	best.medium
	DATA(Response-ConfirmedEventReport(Buf-Scan-Report-*))	Prst	best.medium
	DATA(Invoke-ConfirmedEventReport (MDS-Dynamic-Data-Update-*), ScanReportInfo*)	Prst	best.medium
	DATA(Response-ConfirmedEventReport (MDS-Dynamic-Data-Update-*))	Prst	best.medium

Msg Grp	Message type description	APDU Type	QoS bin type
3	DATA(Invoke-**Un**confirmedAction()): ***<none defined in [IEEE 11073-20601]>***	N/A	N/A
4	DATA(Invoke-ConfirmedAction(MDS-Data-Request), DataRequest)	Prst	best.medium
	DATA(Response-ConfirmedAction(MDS-Data-Request), DataResponse)	Prst	best.medium
	DATA(Invoke-ConfirmedAction(Set-Time), SetTimeInvoke)	Prst	best.medium
	DATA(Response-ConfirmedAction(Set-Time))	Prst	best.medium
	DATA(Invoke-ConfirmedAction(Get-Segment-Info), SegmSelection)	Prst	best.medium
	DATA(Response-ConfirmedAction(Get-Segment-Info), SegmentInfoList)	Prst	best.medium
	DATA(Invoke-ConfirmedAction(Trig-Segment-Data-Xfer), TrigSegmDataXferReq)	Prst	best.medium
	DATA(Response-ConfirmedAction(Trig-Segment-Data-Xfer), TrigSegmDataXferRsp)	Prst	best.medium
	DATA(Invoke-ConfirmedAction(Clear-Segments), SegmSelection)	Prst	best.medium
	DATA(Response-ConfirmedAction(Clear-Segments))	Prst	best.medium
	DATA(Invoke-ConfirmedAction(MDS-Data-Request), DataRequest)	Prst	best.medium
	DATA(Response-ConfirmedAction(MDS-Data-Request), DataResponse)	Prst	best.medium
	DATA(Invoke-ConfirmedAction(MDS-Data-Request), DataRequest)	Prst	best.medium
	DATA(Response-ConfirmedAction(MDS-Data-Request))	Prst	best.medium
5	DATA(Invoke-**Un**confirmedSet()) {scanner OperationalState}	Prst	best.medium
6	DATA(Invoke-ConfirmedSet()){scanner OperationalState}	Prst	best.medium
	DATA(Response-ConfirmSet()){scanner OperationalState}	Prst	best.medium
7	DATA(Invoke-ConfirmedGet()) {MDS attributes}	Prst	best.medium
	DATA(Response-ConfirmGet()){MDS attributes}	Prst	best.medium
	DATA(Invoke-ConfirmedGet()) {PM-Store attributes}	Prst	best.medium
	DATA(Response-ConfirmGet()){PM-Store attributes}	Prst	best.medium
8	DATA(Error(), ErrorResult)	Prst	best.medium
	DATA(Reject()), RejectResult)	Prst	best.medium

6.2.2.6 X73 component - regulatory settings

This clause contains design guidelines that deal with the Continua requirements for regulatory issues using the [IEEE 11073-20601] capabilities. These guidelines are covered in Table 6-13, Table 6-14 and Table 6-15.

For this purpose, the following abstract syntax notation one (ASN.1) definitions are introduced and referenced in Table 6-13.

NOTE: This syntax is also used for Bluetooth LE in clause 11.2.8.

```
ContinuaStructType ::= INT-U8 {
     continua-version-struct(1),   -- auth-body-data is a ContinuaBodyStruct
     continua-reg-struct(2)          -- auth-body-data is a ContinuaRegStruct

}

ContinuaBodyStruct ::= SEQUENCE {
        major-IG-version      INT-U8,
        minor-IG-version      INT-U8,
        certified-capabilities     CertifiedCapabilityClassList
}

CertifiedCapabilityClassList ::= SEQUENCE OF CertifiedCapabilityClassEntry

-- See guideline  11073-20601-CapabilityEntry for the algorithm to compute the
value
CertifiedCapabilityClassEntry ::= INT-U16

ContinuaRegStruct ::= SEQUENCE {
        regulation-bit-field     RegulationBitFieldType
}


RegulationBitFieldType  ::= BITS-16 {
     unregulated-device (0)  -- This bit shall be set if the device is not
regulated }
```

Figure 6-3 – ASN.1 definition of Continua certification structures

6.2.2.6.1 Regulatory / certification information

This clause contains guidelines for the conformance of client components to the usage of regulatory and certification information. The guidelines are contained in Table 6-13.

Table 6-13 – Regulatory / certification information

Name	Description	Comments
11073-20601-Certification	Continua X73-IF service components **shall** support the *Reg-Cert-Data-List* MDS object attribute containing a *RegCertData* element with the *auth-body* field set to *auth-body-continua* and the *auth-body-struc-type* field set to *continua-version-struct* from a ContinuaStructType as defined above. The field *auth-body-data* **shall** be filled in as a *ContinuaBodyStruct* as defined above	Continua certification information - This is used to indicate whether a capability is Continua certified and (if so) which version of the guidelines it is certified to
11073-20601-CapabilitiesList	Continua X73-IF service components **shall** list all implemented and only the implemented Certified Capability Classes in the "certified-capabilities" attribute of the ContinuaBodyStruct structure	
11073-20601-CapabilityEntry	Continua X73-IF service components **shall** assign the following CertifiedCapabilityClassEntry to an implemented Certified Capability Class: MDC_DEV_*_SPEC_PROFILE_* - 4096 + TCode x 8192, where MDC_DEV_*_SPEC_PROFILE_* denotes the IEEE 11073 PHD nomenclature code for the corresponding device (sub-) specialization, and TCode denotes the corresponding transport standard, with TCode = {1 for USB, 2 for Bluetooth BR/EDR, 3 for ZigBee, 4 for Bluetooth LE, and 5 for NFC}. For backward compatibility with CDG version 1 which did not define TCodes, USB and Bluetooth BR/EDR service components **should** additionally include the supported MDC_DEV_*_SPEC_PROFILE_* codes along with a TCode of 0 to interoperate with version 1 client components	Example 1: For a Bluetooth BR/EDR step counter, the assigned CertifiedCapabilityClassEntry computes as 0x4068 (16488 decimal), where it has been substituted MDC_DEV_*_SPEC_PROFILE_* = MDC_DEV_SUB_SPEC_PROFILE_STEP_COUNTER = 4200 and TCode = 2. This gives, 4200 - 4096 + 2 x 8192 = 16488 (0x4068) Example 2: For a ZigBee smoke sensor, the assigned CertifiedCapabilityEntry computes as 0x6077 (24,695 decimal), where it has been substituted MDC_DEV_*_SPEC_PROFILE_* = MDC_DEV_SUB_SPEC_PROFILE_SMOKE_SENSOR = 4215 and TCode = 3. This gives, 4215 – 4096 + 3 x 8192 = 24,695 (0x6077)

Name	Description	Comments
11073-20601-DeviceSpecList	Continua X73-IF service components **shall** list MDC_DEV_SPEC_PROFILE_* value(s) corresponding to each supported Continua certified Capability Class in the System-Type-Spec-List attribute of the MDS object. The attribute **may** contain additional MDC_DEV_SPEC_PROFILE_* value(s) corresponding to supported IEEE specializations that are not Continua certified	
11073-20601-Regulation	Continua X73-IF service components **shall** support the *Reg-Cert-Data-List* MDS object attribute containing a *RegCertData* element with the *auth-body* field set to *auth-body-continua* and the *auth-body-struc-type* field set to *continua-reg-struct* from a ContinuaStructType as defined below. The field *auth-body-data* **shall** be filled in as a *ContinuaRegStruct* as defined below	Regulation Information - This is used to provide a coarse regulatory indication (e.g. "Regulated or Not Regulated")

6.2.2.6.2 Conformance

This clause contains guidelines for the conformance of client components to [IEEE 11073-20601] and [ISO/IEEE 11073-104xx] specifications and capabilities. The guidelines are contained in Table 6-14.

Table 6-14 – Manager conformance

Name	Description	Comments
11073-20601-Manager-Conformance	Continua X73-IF client components **shall** appropriately utilize the mandatory measurement objects from compliant device specializations	In the context of these requirements, the term "appropriately utilize" implies that the objects get utilized in accordance with the function of the device. That is, a mandatory measurement object can be displayed, and/or forwarded, and/or used as input for an assessment algorithm, etc.

Name	Description	Comments
11073-20601-Utilization-Documentation	Continua X73-IF client components **shall** provide to the Test and Certification organization documentation on the appropriate utilization of the individual mandatory measurement objects	

6.2.2.6.3 Nomenclature codes

This clause contains guidelines for the use of nomenclature codes by client and service components. The guidelines are contained in Table 6-15.

Table 6-15 – Nomenclature codes

Name	Description	Comments
11073-20601-Continua-Nomenclature-Codes	Continua X73-IF service and client components that use private nomenclature codes **shall** allocate them from the range 0xF000 through 0xFBFF	The range from 0xFC00 through 0xFFFF is reserved for future use by the CDG

6.2.2.7 X73 component – user identification

This clause contains guidelines for service components on user identification. The guidelines are contained in Table 6-16.

Table 6-16 – User identification

Name	Description	Comments
11073-20601-PID-ScanReport	Continua X73-IF service components designed to store and utilize data from multiple users simultaneously and that use agent-initiated measurement data transmission **shall** identify users and set the person-id field in the corresponding ScanReportPer* structure	Identification means distinguishing between users of the measurement device
11073-20601-PID-PM-Store	Continua X73-IF service components designed to store and utilize data from multiple users simultaneously in one or more PM-stores **shall** identify users and support the PM-Seg-Person-Id PM-segment object attribute and set the pmsc-multi-person bit in the PM-Store-Capab PM-Store object attribute	Identification means distinguishing between users of the measurement device

6.2.3 Standard configuration support

This paragraph contains guidelines on the support of standard and extended configurations by X73-IF client and service components to better guarantee interoperability. The guidelines are covered by Table 6-17.

Table 6-17 – Communication capabilities – general

Name	Description	Comments
11073-20601-standard-config-support	Continua X73-IF service components **shall** (always) support one of the pre-defined standard configurations for supported [ISO/IEEE 11073-104xx] device specializations if such configurations are defined in the corresponding [ISO/IEEE 11073-104xx] device specialization.	[ISO/IEEE 11073-20601-2014] and later no longer requires a service component to always support a standard configuration: a service component that supports an extended configuration with a PM store does not need to support a standard configuration. The CDGs do require support for a standard configuration by the service component to maintain interoperability.
11073-20601-extended-config-support	Continua X73-IF client components that support protocol IEEE 11073-20601 protocol v3 **should** support extended configurations as used by [ISO/IEEE 11073-104xx] device implementations for supported [ISO/IEEE 11073-104xx] device specializations as long as these configurations consist of objects and attributes defined in the corresponding [ISO/IEEE 11073-104xx] specification.	[IEEE 11073-20601-2014] and later no longer require a service component to always support a standard configuration: a service component that supports an extended configuration with a PM store does not need to support a standard configuration. The CDGs require that such extended configurations should be supported by the PHG for improved interoperability.

Note: the following device specializations do not define standard configurations:

- 11073-10441 Cardiovascular fitness and activity monitor
- 11073-10442 Strength fitness equipment
- 11073-10471 Independent living activity hub

6.2.4 Sensor component – communication capabilities

This clause contains guidelines for general communications capabilities of sensor components.

Table 6-18 – Communication capabilities association and configuration

Name	Description	Comments
11073-20601-Complete-Config-Object-List	Continua X73-IF service components **shall** always populate the ConfigObjectList of a configuration message with the complete set of objects and attributes supported by the configuration	[IEEE 11073-20601] allows an agent to send a configuration event with an empty ConfigObjectList if the configuration-id is within the range of standard-config-start and standard-config-end. This mechanism was designed in [IEEE 11073-20601] to optimize bytes transferred. However this mechanism is likely to cause interoperability problems as the feature is not well known. It is believed that the enhancement to interoperability outweighs the optimization.

6.2.5 Sensor component multi-function devices

This clause describes guidelines for multi-function devices (e.g. how to make combined use of [ISO/IEEE 11073-104xx] to create multi-function devices, or how to use the [IEEE 11073-20601] mechanisms for association in this case).

Table 6-19 – Multi-function devices

Name	Description	Comments
11073-20601-Multi-Function	A Continua X73-IF service component **shall** have at most one [IEEE 11073-20601] association to an X73-IF client component at any point in time regardless of whether the device is a single function or multi-function device	This guideline prohibits the device from having two concurrent associations. The device may provide different configuration options only in subsequent associations only after closing the currently active association

6.3 X73 Devices

This clause contains guidelines for client and service components that implement specific [ISO/IEEE 11073-104xx] device specializations. There is a subclause per device specialization.

6.3.1 Pulse oximeter

This clause contains guidelines for client and service components that implement the pulse oximeter device specialization. The guidelines are contained in Table 6-20 and additionally in Table 6-21 and Table 6-22 for implementations supporting PM stores.

6.3.1.1 Pulse oximeter – general requirements

Table 6-20 – Pulse oximeter – general requirements

Name	Description	Comments
11073-10404-Reqt	Continua X73 pulse oximeter service and client components **shall** implement [ISO/IEEE 11073-10404]	
11073-Pulse-Oximeter-PM-Store	Continua X73 pulse oximeter service and client components that implement and use the PM-Store model **shall** implement the guidelines in Table 6-21, and Table 6-22 as well as Table 6-2 or Table 6-3 and subsequent explanatory text.	

6.3.1.2 PM-store objects for the pulse oximeter

The PM-store and PM-segment classes provide a flexible and powerful means for storing large amounts of measurement data for later transmission to a PHG. However, this flexibility could potentially lead to ambiguities that could jeopardize interoperability. This clause describes recommended implementations for the most common use case, the sleep study.

Figure 6-4 illustrates one arrangement of a PM-store organized into two PM-segments. Each PM-Segment stores periodically sampled data from a single contiguous session, and each PM-segment entry contains a SpO_2 measurement and a pulse rate measurement sampled at a single point in time.

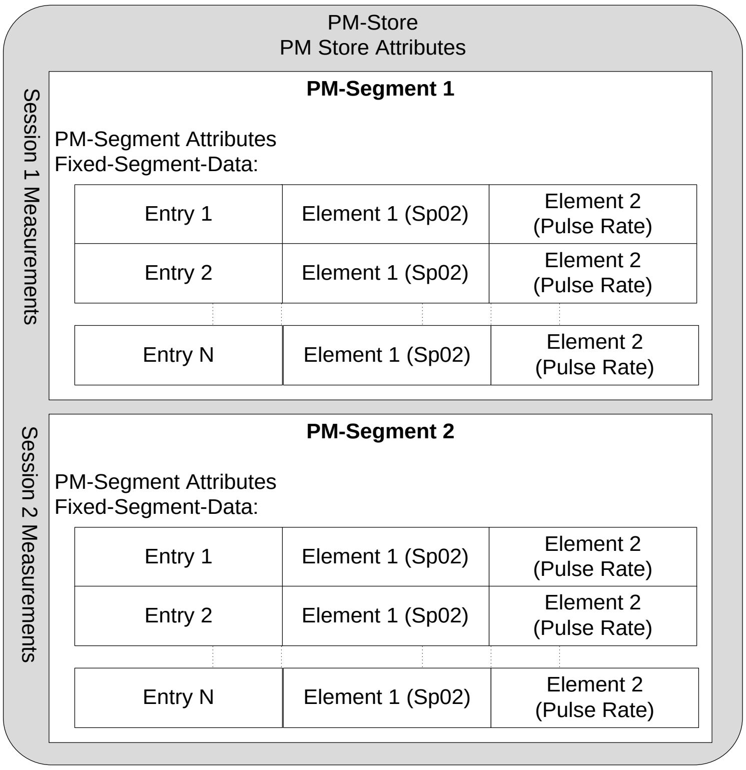

Figure 6-4 – PM-Store usage for pulse oximeter

Some situations may not be suitable for the previous recommendation. For instance, a pulse oximeter may record SpO_2 measurements at a different sampling period than pulse rate measurements, or one of the measurements during a session could conceivably be episodic. A PM-segment organization that could be better suited to this situation is illustrated in Figure 6-5.

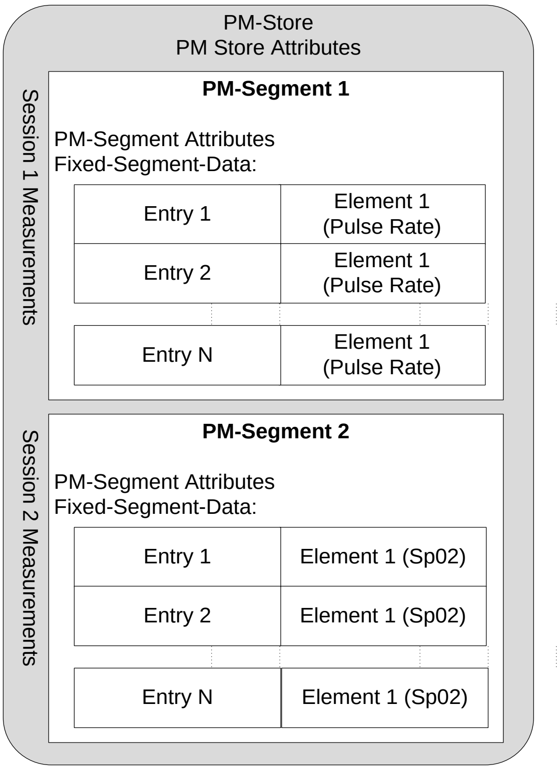

Figure 6-5 – Alternate PM-Segment organization

This alternate arrangement challenges the notion of measurement association. Given a collection of PM-segments, how can the PHG determine which, if any, segments are associated?
Time stamps are used to determine whether one or more PM-segments are associated with another. Any measurements within one or more PM-segments in a PM-store are considered to be associated if their start and end segment attributes are overlapping, or if one segment's time range is contained within another segment. Table 6-21 prohibits the storage of associated PM-segments in separate PM-stores, which would add unnecessary complexity for client components to identify associated PM-segments.

Table 6-21 – Pulse Oximeter PM-Store measurement requirements

Name	Description	Comments
11073-Pulse-Oximeter-PM-Store-Organization	Continua X73 pulse oximeter service components **should** organize their stored measurements as shown in Figure 6-4 or Figure 6-5	The order of SpO2 and pulse rate is defined in the SegEntryMap

Name	Description	Comments
11073-Pulse-Oximeter-PM-Store-StartTime-StopTime	Continua X73 pulse oximeter service components **shall** store the start time and end time in the PM-Segment attributes *Segment-Start-Abs-Time* and *Segment-end-Abs-Time*	Enables the PHG to determine whether one or more PM-segments are associated
11073-Pulse-Oximeter-PM-Store-Associated-Measurements-Locations	Continua X73 pulse oximeter service components **shall** create PM-segments within the same PM-store, if the PM-segments are overlapping in time	PM-segments are considered to be overlapping in time if the time ranges defined by their *Segment-Start-Abs-Time* and *Segment-End-Abs-Time* attribute values are overlapping

6.3.1.3 PM-Store Object Attributes

Table 6-22 – Pulse Oximeter PM-Store object attributes guideline

Name	Description	Comments
11073-Pulse-Oximeter-PM-Store-Object-Attributes-PM-Store-Capab-set	Continua X73 pulse oximeter service components **shall** set the following bit value for the PM-store-Capab attribute of the PM-store Object: *pmsc-clear-segm-by-all-sup*	
11073-Pulse-Oximeter-PM-Store-Object-Attributes-PM-Store-Capab-clear	Continua X73 pulse oximeter service components **shall** clear the following bit value for the PM-store-Capab attribute of the PM-Store object: *pmsc-clear-segm-by-time-sup*	
11073-Pulse-Oximeter-PM-Store-Object-Attributes-PM-Store-Label	Continua X73 pulse oximeter service components, that implement the PM-store-Label attribute of the PM-store object, **shall** not set a value of size larger than 255 octets	
11073-Pulse-Oximeter-PM-Store-Object-Attributes-Sample-Period-Attribute	Continua X73 pulse oximeter service components **shall** implement the *Sample-Period* attribute of a PM-store object, if the stored measurements are periodic and the *Sample-Period* attribute is not implemented in each of the PM-segment objects created within that PM-store object. If the Sample-Period is defined in both the PM-store and in the PM-segment(s), the PM-segment attribute value **shall** take precedence	

Name	Description	Comments
11073-Pulse-Oximeter-PM-Store-Object-alignment	Continua X73 pulse oximeter service components **shall** align periodic measurements so that the time of the first measurement is equivalent to *Segment-Start-Abs-Time*	Need to align events in case two associated PM-segments have widely varying sample periods

6.3.2 Basic 1-3 lead ECG

This clause contains guidelines for client and service components that implement the ECG device specialization. The guidelines are contained in Table 6-23 and additionally in Table 6-24 and Table 6-24 for implementations supporting PM stores.

Table 6-23 – Basic 1-3 lead ECG – general requirements

Name	Description	Comments
11073-10406-Basic-ECG-Reqt	Continua X73 Basic 1-3 lead ECG service and client components **shall** implement [IEEE 11073-10406]	
11073-10406-Simple-ECG-Profile	Continua X73 Basic 1-3 lead ECG service and client components **shall** implement the simple ECG profile defined in [IEEE 11073-10406]	The simple ECG profile defined in [IEEE 11073-10406] mandates implementation of ECG waveform functionality
11073-Basic-ECG-PM-Store	Continua X73 Basic 1-3 lead ECG service and client components that implement and use the PM-Store model **shall** implement the guidelines in Table 6-24 and Table 6-25, and **should** follow the storage layout as shown in Figure 7 of [IEEE 11073-10406]	Figure 7 of [IEEE 11073-10406] illustrates the example of a 3-lead Basic 1-3 lead ECG, with measurement data from all leads being contained in each entry preceded by a segment entry header. For a lower number of leads the number of elements in each entry reduces accordingly. The order of elements within an entry is defined in the SegEntryMap attribute

6.3.2.1 PM-store objects for the Basic 1-3 lead ECG

The PM-store and PM-segment classes provide a flexible and powerful means for storing large amounts of measurement data for later transmission to a PHG. However, this flexibility could potentially lead to ambiguities that could jeopardize interoperability. This clause describes recommended implementations for the most common use case involving persistently stored metric data, the storage of ECG waveform data.

Figure 7 of [IEEE 11073-10406] illustrates one arrangement of a periodic PM-store organized into two PM-segments. Each PM-Segment stores periodically sampled data from a single contiguous session, and each PM-segment entry contains sample arrays of ECG waveform data for all implemented leads sampled during the same period of time.

Some situations may not be suitable for the previous recommendation. For instance, a Basic 1-3 lead ECG may record heart-rate measurements at a different sampling period than ECG waveform measurements, or one of the measurements during a session could conceivably be aperiodic. A PM-segment organization that could be better suited to this situation is to use a separate PM-segment for different measurement types. See also Figure 6-5 for a conceptual illustration of this type of PM-segment organization. This alternate arrangement challenges the notion of measurement association, i.e., for the PHG to determine which segments are associated for a given collection of PM-segments. Storage of periodic and aperiodic measurements involves organization in separate aperiodic and periodic PM-stores, respectively.

Time stamps are used to determine whether one or more PM-segments are associated with another. Any measurements within one or more PM-segments in a PM-store are considered to be associated if their start and end segment attributes are overlapping, or if one segment's time range is contained within another segment. Table 6-24 prohibits the storage of associated PM-segments in separate PM-stores, which would add unnecessary complexity for client components to identify associated PM-segments.

Table 6-24 – ECG PM-Store measurement requirements

Name	**Description**	**Comments**
11073-Basic-ECG-Periodic-PM-Store-Associated-Measurements-Locations	For periodic measurements, Continua X73 Basic 1-3 lead ECG service components **shall** create PM-segments within the same periodic PM-store, if the PM-segments are overlapping in time	PM-segments are considered to be overlapping in time if the time ranges defined by their *Segment-Start-Abs-Time* and *Segment-End-Abs-Time* attribute values are overlapping
11073-Basic-ECG-Aperiodic-PM-Store-Associated-Measurements-Locations	For aperiodic measurements, Continua X73 Basic 1-3 lead ECG service components **shall** create PM-segments within the same aperiodic PM-store, if the PM-segments are overlapping in time	PM-segments are considered to be overlapping in time if the time ranges defined by their *Segment-Start-Abs-Time* and *Segment-End-Abs-Time* attribute values are overlapping

6.3.2.2 PM-Store object attributes

Table 6-25 – ECG PM-Store object attributes guidelines

Name	**Description**	**Comments**
11073-Basic-ECG-PM-Store-Object-Attributes-PM-Store-Label	Continua X73 Basic 1-3 lead ECG service components, that implement the PM-Store-Label attribute of the PM-Store object, **shall** not set a value of size larger than 255 octets	
11073-Basic-ECG-PM-Store-Object-alignment	Continua X73 Basic 1-3 lead ECG service components **shall** align periodic measurements such that the time of the first measurement is equivalent to *Segment-Start-Abs-Time*	Need to align events in case two associated PM-segments have widely varying sample periods

6.3.3 Heart-rate sensor

This clause contains guidelines for client and service components that implement the heart-rate sensor device specialization. The guidelines are contained in Table 6-26 and additionally in Table 6-27 and Table 6-28 for implementations supporting PM stores.

Table 6-26 – Heart-rate sensor – general requirements

Name	Description	Comments
11073-10406-Heart-Rate-Reqt	Continua X73 heart-rate sensor service and client components **shall** implement [IEEE 11073-10406]	
11073-10406-Heart-Rate-Profile	Continua X73 heart-rate sensor service and client components **shall** implement the heart rate profile defined in [IEEE 11073-10406]	The heart rate profile defined in [IEEE 11073-10406] mandates the implementation of heart-rate functionality
11073-Heart-Rate-PM-Store	Continua X73 heart-rate sensor service and client components that implement and use the PM-Store model **shall** implement the guidelines in Table 6-27 and Table 6-28	For simple heart-rate sensors PM-Store functionality is typically not implemented. This guideline provides guidance for the case that PM-Store functionality is implemented

6.3.3.1 PM-store objects for the heart-rate sensor

The PM-store and PM-segment classes provide a flexible and powerful means for storing large amounts of measurement data for later transmission to a PHG. For simple heart-rate sensors this functionality is typically not implemented. However, if implemented this clause provides guidance to ensure interoperability.

A common use case involves persistently stored R-R interval data. Figure 6-6 illustrates a simple arrangement of an aperiodic PM-store containing PM-segments for storing R-R interval data from different measurement sessions. The entries of a PM-segment each contain an element of R-R interval data.

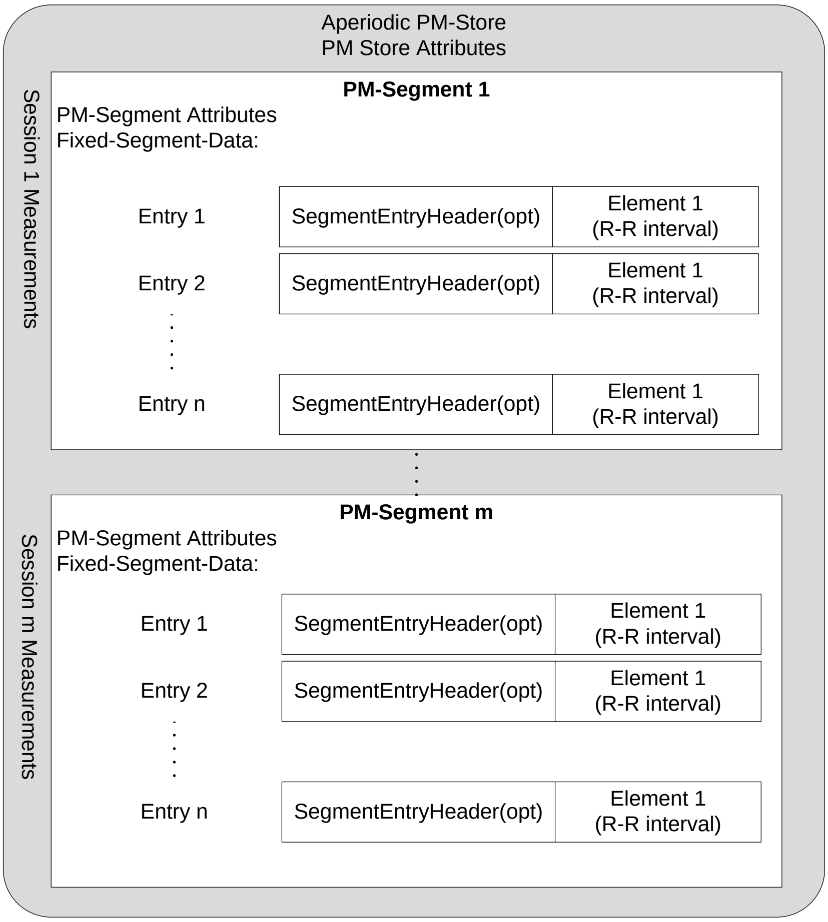

Figure 6-6 – PM-store usage example for heart-rate sensor

Time stamps are used to determine whether one or more PM-segments are associated with another. Any measurements within one or more PM-segments in a PM-store are considered to be associated if their start and end segment attributes are overlapping, or if one segment's time range is contained within another segment. Table 6-27 prohibits the storage of associated PM-segments in separate PM-stores, which would add unnecessary complexity for client components to identify associated PM-segments.

Table 6-27 – Heart-rate sensor PM-store measurement requirements

Name	Description	Comments
11073-Heart-rate-Periodic-PM-Store-Associated-Measurements-Locations	For periodic measurements, Continua X73 heart-rate sensor service components **shall** create PM-segments within the same periodic PM-store, if the PM-segments are overlapping in time	PM-segments are considered to be overlapping in time if the time ranges defined by their *Segment-Start-Abs-Time* and *Segment-End-Abs-Time* attribute values are overlapping
11073-Heart-Rate-Aperiodic-PM-Store-Associated-Measurements-Locations	For aperiodic measurements, Continua X73 heart-rate sensor service components **shall** create PM-segments within the same aperiodic PM-store, if the PM-segments are overlapping in time	PM-segments are considered to be overlapping in time if the time ranges defined by their *Segment-Start-Abs-Time* and *Segment-End-Abs-Time* attribute values are overlapping

6.3.3.2 PM-Store object attributes

Table 6-28 – Heart-rate sensor PM-Store object attributes guidelines

Name	Description	Comments
11073-Heart-Rate-PM-Store-Object-Attributes-PM-Store-Label	Continua X73 heart-rate sensor service components, that implement the PM-Store-Label attribute of the PM-Store object, **shall** not set a value of size larger than 255 octets	
11073-Heart-Rate-PM-Store-Object-alignment	Continua X73 heart-rate sensor service components **shall** align periodic measurements such that the time of the first measurement is equivalent to *Segment-Start-Abs-Time*	Need to align events in case two associated PM-segments have widely varying sample periods

6.3.4 Blood pressure monitor

This clause contains guidelines for client and service components that implement the blood pressure monitor device specialization. The guidelines are contained in Table 6-29.

Table 6-29 – Blood pressure monitor – general requirements

Name	Description	Comments
11073-10407-Reqt	Continua X73 blood pressure monitor service and client components **shall** implement [ISO/IEEE 11073-10407]	

6.3.5 Thermometer

This clause contains guidelines for client and service components that implement the thermometer device specialization. The guidelines are contained in Table 6-30.

Table 6-30 – Thermometer – general requirements

Name	Description	Comments
11073-10408-Reqt	Continua X73 thermometer service and client components **shall** implement [ISO/IEEE 11073-10408]	

6.3.6 Weighing-scales

This clause contains guidelines for client and service components that implement the weighing scales device specialization. The guidelines are contained in table Table 6-31.

Table 6-31 – Weighing-scales – general requirements

Name	Description	Comments
11073-10415-Reqt	Continua X73 weighing-scales service and client components **shall** implement [ISO/IEEE 11073-10415]	

6.3.7 Glucose Meter

This clause contains guidelines for client and service components that implement the glucose meter device specialization. The guidelines are contained in table Table 6-32.

Table 6-32 – Glucose Meter General Requirements

Name	Description	Comments
11073-10417-Reqt	Continua X73 Glucose Meter service and client components **shall** implement [IEEE 11073-10417]	

6.3.8 INR meter

This clause contains guidelines for client and service components that implement the INR meter device specialization. The guidelines are contained in table Table 6-33.

Table 6-33 – INR meter – general requirements

Name	Description	Comments
11073-10418-Reqt	Continua X73 INR meter service and client components **shall** implement [IEEE 11073-10418]	

6.3.9 Body composition analyzer

This clause contains guidelines for client and service components that implement the body composition analyzer device specialization. The guidelines are contained in table Table 6-34.

Table 6-34 – Body composition analyzer general requirements

Name	Description	Comments
11073-10420-Reqt	Continua X73 Body composition analyzer service and client components **shall** implement [IEEE 11073-10420]	

6.3.10 Peak flow monitor

This clause contains guidelines for client and service components that implement the peak flow monitor device specialization. The guidelines are contained in Table 6-35.

Table 6-35 – Peak flow monitor – general requirements

Name	Description	Comments
11073-10421-Reqt	Continua X73 peak flow monitor service and client components **shall** implement [ISO/IEEE 11073-10421]	

6.3.11 Cardiovascular fitness

This clause contains guidelines for client and service components that implement the cardiovascular fitness device specialization. The guidelines are contained in Table 6-36.

Table 6-36 – Cardiovascular fitness – general requirements

Name	Description	Comments
11073-10441-Reqt	Continua X73 cardiovascular fitness service and client components **shall** implement [IEEE 11073-10441]	

6.3.12 Cardiovascular step counter

There is no IEEE 11073 device specialization dedicated to a cardiovascular step counter. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10441] to create an X73 cardiovascular step counter. The guidelines are contained in Table 6-37.

Table 6-37 – Cardiovascular step counter – general requirements

Name	Description	Comments
11073-10441-Reqt	Continua X73 cardiovascular step counter service and client components **shall** implement [IEEE 11073-10441]	
11073-Step-Counter-Service-Max-APDU	Continua X73 cardiovascular step counter service components **shall** be able to support a maximum APDU size of 224 octets from Continua X73-IF client components	These are consistent with weighing scale, thermometer, glucose meter, blood pressure monitor, and independent living activity hub
11073-Step-Counter-Client-Max-APDU	Continua X73 cardiovascular step counter client components **shall** be able to support a maximum APDU size of 6624 octets from Continua X73-IF service components	
11073-Step-Counter-Service-Mandatory-Objects	Continua X73 cardiovascular step counter service components **shall** support the session and distance object in units of steps	

Name	Description	Comments
11073-Step-Counter-Client-Mandatory-Objects	Continua X73 cardiovascular step counter client components **shall** support the session and distance object (all unit codes)	
11073-Step-Counter-Service-Optional-Objects	Continua X73 cardiovascular step counter service components **may** support the subsession, cadence, speed, distance (in meters and/or feet), stride length, or energy expended objects as defined in [IEEE 11073-10441]	
11073-Step-Counter-Client-Optional-Objects	Continua X73 cardiovascular step counter client components **may** support the subsession, cadence, speed, stride length, or energy expended objects as defined in [ISO/IEEE 11073-10441]	
11073-Step-Counter-MDC-Code	Continua X73 step counter service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_STEP_COUNTER = 4200 (0x1068)	

6.3.13 Strength fitness

This clause contains guidelines for client and service components that implement the strength fitness device specialization. The guidelines are contained in Table 6-38.

Table 6-38 – Strength fitness – general requirements

Name	Description	Comments
11073-10442-Reqt	Continua X73 strength fitness service and client components **shall** implement [ISO/IEEE 11073-10442]	

6.3.14 Activity hub

This clause contains guidelines for client and service components that implement the activity hub device specialization. The guidelines are contained in Table 6-39.

Table 6-39 – Activity hub – general requirements

Name	Description	Comments
11073-10471-Reqt	Continua X73 activity hub service and client components **shall** implement [ISO/IEEE 11073-10471]	

6.3.15 Fall sensor

There is no IEEE 11073 device specialization dedicated to a fall sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create a PHD fall sensor. The guidelines are covered by Table 6-40.

Table 6-40 – Fall sensor – general requirements

Name	Description	Comments
11073-10471-Fall-Reqt	Continua X73 fall sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Fall-Sensor-Object	Continua X73 fall sensor service and client components **shall** implement the fall sensor enumeration object	
11073-Fall-Sensor-MDC-Code	Continua X73 fall sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_FALL_SENSOR = 4213 (0x1075)	

6.3.16 Motion sensor

There is no IEEE 11073 device specialization dedicated to a motion sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create a PHD motion sensor. The guidelines are covered by Table 6-41.

Table 6-41 – Motion sensor – general requirements

Name	Description	Comments
11073-10471-Motion-Reqt	Continua X73 motion sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Motion-Sensor-Object	Continua X73 motion sensor service and client components **shall** implement the motion sensor enumeration object	
11073-Motion-Sensor-MDC-Code	Continua X73 motion sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_MOTION_SENSOR = 4219 (0x107B)	

6.3.17 Enuresis sensor

There is no IEEE 11073 device specialization dedicated to an enuresis sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 enuresis sensor. The guidelines are covered by Table 6-42.

Table 6-42 – Enuresis sensor – general requirements

Name	Description	Comments
11073-10471-Enuresis-Reqt	Continua X73 enuresis sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Enuresis-Sensor-Object	Continua X73 enuresis sensor service and client components **shall** implement the enuresis sensor enumeration object	
11073-Enuresis-Sensor-MDC-Code	Continua X73 enuresis sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_ENURESIS_SENSOR = 4221 (0x107D)	

6.3.18 Contact closure sensor

There is no IEEE 11073 device specialization dedicated to a contact closure sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 contact closure sensor. The guidelines are covered by Table 6-43.

Table 6-43 – Contact closure sensor – general requirements

Name	Description	Comments
11073-10471-Contact-Reqt	Continua X73 contact closure sensor service and client components **shall** implement ISO/IEEE 11073-10471-2008	
11073-Contact-Closure-Sensor-Object	Continua X73 contact closure sensor service and client components **shall** implement the contact closure sensor enumeration object	
11073-Contact-Closure-Sensor-MDC-Code	Continua X73 contact closure sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_CONTACTCLOSURE_SENSOR = 4222 (0x107E)	

6.3.19 Switch sensor

There is no IEEE 11073 device specialization dedicated to a switch use sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 switch sensor. The guidelines are covered by Table 6-44.

Table 6-44 – Switch use sensor – general requirements

Name	Description	Comments
11073-10471-Switch-Reqt	Continua X73 switch sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Switch-Sensor-Object	Continua X73 switch sensor service and client components **shall** implement the Switch use sensor enumeration object	
11073-Switch-Sensor-MDC-Code	Continua X73 switch sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_SWITCH_SENSOR = 4224 (0x1080)	

6.3.20 Dosage sensor

There is no IEEE 11073 device specialization dedicated to a medication dosage sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 dosage sensor. The guidelines are covered by Table 6-45.

Table 6-45 – Dosage sensor – general requirements

Name	Description	Comments
11073-10471-Dosage-Reqt	Continua X73 dosage sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Dosage-Sensor-Object	Continua X73 dosage sensor service and client components **shall** implement the medication dosage sensor enumeration object	
11073-Dosage-Sensor-MDC-Code	Continua X73 dosage sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_DOSAGE_SENSOR = 4225 (0x1081)	

6.3.21 Water sensor

There is no IEEE 11073 device specialization dedicated to a water sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 water sensor. The guidelines are covered by Table 6-46.

Table 6-46 – Water sensor – general requirements

Name	Description	Comments
11073-10471-Water-Reqt	Continua X73 Water Sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Water-Sensor-Object	Continua X73 water sensor service and client components **shall** implement the water sensor enumeration object	
11073-Water-Sensor-MDC-Code	Continua X73 water sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_WATER_SENSOR = 4217 (0x1079)	

6.3.22 Smoke sensor

There is no IEEE 11073 device specialization dedicated to a smoke sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 smoke sensor. The guidelines are covered by Table 6-47.

Table 6-47 – Smoke sensor – general requirements

Name	Description	Comments
11073-10471-Smoke-Reqt	Continua X73 smoke sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Smoke-Sensor-Object	Continua X73 smoke sensor service and client components **shall** implement the smoke sensor enumeration object	
11073-Smoke-Sensor-MDC-Code	Continua X73 smoke sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_SMOKE_SENSOR = 4215 (0x1077)	

6.3.23 Property exit sensor

There is no IEEE 11073 device specialization dedicated to a property exit sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 property exit sensor. The guidelines are covered by Table 6-48.

Table 6-48 – Property exit sensor – general requirements

Name	Description	Comments
11073-10471-Exit-Reqt	Continua X73 property exit sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Property-Exit-Sensor-Object	Continua X73 property exit sensor service and client components **shall** implement the property exit sensor enumeration object	
11073-Property-Exit-Sensor-MDC-Code	Continua X73 property exit sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_PROPEXIT_SENSOR = 4220 (0x107C)	

6.3.24 Temperature sensor

There is no IEEE 11073 device specialization dedicated to a temperature sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 temperature sensor. The guidelines are covered in Table 6-49.

Table 6-49 – Temperature sensor – general requirements

Name	Description	Comments
11073-10471-Temperature-Reqt	Continua X73 temperature sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Temperature-Sensor-Object	Continua X73 temperature sensor service and client components **shall** implement the temperature sensor enumeration object	
11073-Temperature-Sensor-MDC-Code	Continua X73 temperature sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_TEMP_SENSOR = 4226 (0x1082)	

6.3.25 Usage sensor

There is no IEEE 11073 device specialization dedicated to a usage sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 usage sensor. The guidelines are covered in Table 6-50.

Table 6-50 – Usage sensor – general requirements

Name	Description	Comments
11073-10471-Usage-Reqt	Continua X73 usage sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Usage-Sensor-Object	Continua X73 usage sensor service and client components **shall** implement the usage sensor enumeration object	
11073-Usage-Sensor-MDC-Code	Continua X73 usage sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_USAGE_SENSOR = 4223 (0x107F)	

6.3.26 PERS sensor

There is no IEEE 11073 device specialization dedicated to a PERS sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 PERS sensor. The guidelines are covered in Table 6-51.

Table 6-51 – PERS sensor – general requirements

Name	Description	Comments
11073-10471-PERS-Reqt	Continua X73 PERS sensor service and client components **shall** implement ISO/IEEE 11073-10471-2008	
11073-PERS-Sensor-Object	Continua X73 PERS sensor service and client components **shall** implement the PERS sensor enumeration object	
11073-PERS-Sensor-MDC-Code	Continua X73 PERS sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_PERS_SENSOR = 4214 (0x1076)	

6.3.27 CO sensor

There is no IEEE 11073 device specialization dedicated to a carbon monoxide sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 CO sensor. The guidelines are covered in Table 6-52.

Table 6-52 – CO sensor – general requirements

Name	Description	Comments
11073-10471-CO-Reqt	Continua X73 CO Sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-CO-Sensor-Object	Continua X73 CO sensor service and client components **shall** implement the CO sensor enumeration object	
11073-CO-Sensor-MDC-Code	Continua X73 CO sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_FALL_SENSOR = 4216 (0x1078)	

6.3.28 Gas sensor

There is no IEEE 11073 device specialization dedicated to a gas sensor. This clause gives guidelines how to make use of the generic functionality of [ISO/IEEE 11073-10471] to create an X73 gas sensor. The guidelines are covered by Table 6-53.

Table 6-53 – Gas sensor – general requirements

Name	Description	Comments
11073-10471-Gas-Reqt	Continua X73 gas sensor service and client components **shall** implement [ISO/IEEE 11073-10471]	
11073-Gas-Sensor-Object	Continua X73 gas sensor service and client components **shall** implement the gas sensor enumeration object	
11073-Gas-Sensor-MDC-Code	Continua X73 gas sensor service components **shall** set the MDC_DEV_*_SPEC_PROFILE_* code to MDC_DEV_SUB_SPEC_PROFILE_GAS_SENSOR = 4218 (0x107A)	

6.3.29 Adherence monitor

This clause contains guidelines for client and service components that implement the adherence monitor device specialization. The guidelines are contained in Table 6-54.

Table 6-54 – Adherence monitor – general requirements

Name	Description	Comments
11073-10472-Reqt	Continua X73 adherence monitor service and client components **shall** implement [IEEE 11073-10472]	

6.3.30 Sleep Apnea Breathing Therapy Equipment (SABTE)

This clause contains guidelines for client and service components that implement the SABTE device specialization. The guidelines are contained in Table 6-55.

Table 6-55 – SABTE – General Requirements

Name	Description	Comments
11073-10424-Reqt	Continua X73 SABTE service and client components **shall** implement [IEEE 11073-10424]	

6.3.31 Continuous Glucose Monitor (CGM)

This clause contains guidelines for client and service components that implement the CGM device specialization. The guidelines are contained in Table 6-56.

Table 6-56 – Continuous Glucose Monitor General Requirements

Name	Description	Comments
11073-10425-Reqt	Continua X73 CGM service and client components **shall** implement [IEEE 11073-10425]	

6.3.32 Insulin Pump (IP)

This clause contains guidelines for client and service components that implement the insulin pump device specialization. The guidelines are contained in Table 6-57.

Table 6-57 – Insulin Pump General Requirements

Name	Description	Comments
11073-10419-Reqt	Continua X73 IP service and client components **shall** implement [IEEE 11073-10419]	

7 NFC interface design guidelines

7.1 NFC interface architecture (informative)

This clause lists the design guidelines specific for interoperability of Personal Health Devices and Personal Health Gateways using the NFC interface. The position of the NFC interface in the Continua architecture is illustrated in Figure 7-1.

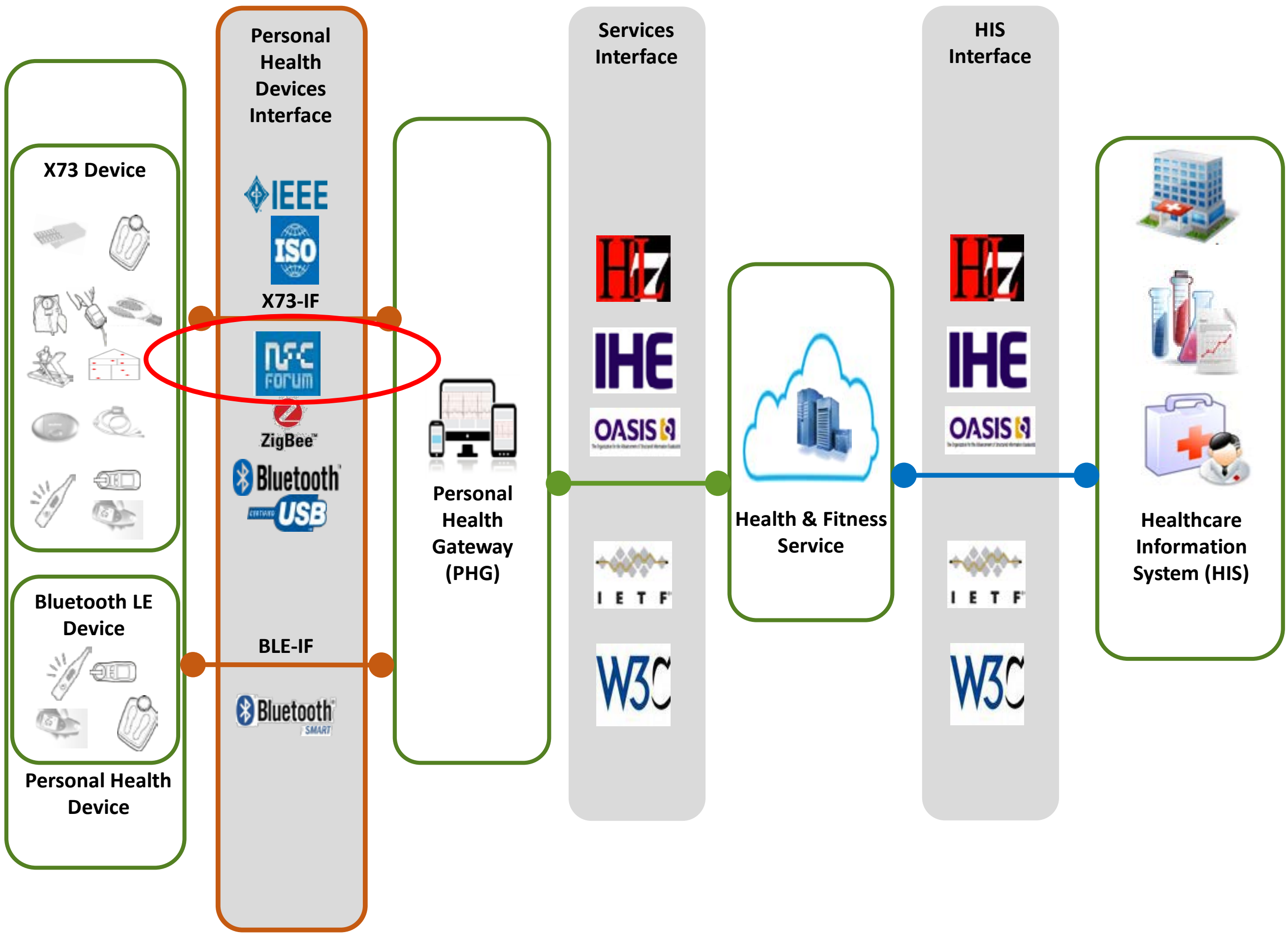

Figure 7-1 – NFC interface context

7.1.1 Overview

NFC enables a Continua Personal Health Device (PHD) to communicate with a Continua Personal Health Gateway (PHG) by touch. A user brings the two devices into close proximity for a short period of time – typically by touching one device with the other. While the devices are touching, data may be exchanged bidirectionally. In a typical use case a user would transfer blood pressure readings from their blood pressure meter (Continua PHD) to a mobile phone (Continua PHG) by simply touching the two devices together.

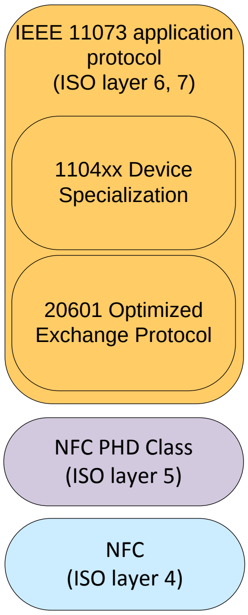

Figure 7-2 – NFC interface stack

7.1.2 Transport protocols and selected standards

[NFC PHDC] has been selected to serve as the transports for the NFC interface (NFC-IF)
The selected protocol for the transport layer ensures interoperable set-up and tear-down of the communication channel for the transfer of control and data messages across all domains. Note that NFC works over a range of up to 10 centimeters, so that actual touching of devices might not even be required.

7.1.3 Exchange protocols and selected standards

For the data and messaging layer of the NFC-IF, the IEEE 11073 Personal Health Device family of standards has been selected. For the detailed list of selected data/messaging layer standards; please see Clause 6.

7.1.4 Device communication styles

NFC is intended for a batch communication style. This style requires the transport between the device and the PHG to communicate previously collected data points at a later time. The user chooses the moment of communication by touching the devices.
In QoS terms explained in Clause 6.1.6 of [H.810], NFC is best.medium. Communication is acknowledged and must be complete or the transaction is rejected. Latency is typically <1 second for a NFC application.

7.1.5 NFC interface security

For a NFC solution, it is assumed that the physical action of the user touching two devices provides a level of security to prevent too easy inadvertent leakage of data to a different PHG.
Designers of NFC PHDs should take normal care for NFC systems to ensure a robust design that cannot be easily intercepted or interrogated by an antenna that is not in very close physical contact – touching. Typically this is done by managing power and physically shielding components to ensure that only two antennas that are in very close contact are capable of communication exchange.
Note that such measures help to increase the security of the system, but they cannot prevent the effects of all security threats that are inherent to the nature of NFC. It is advised that PHD

manufacturers implement suitable security controls and mechanisms based on a security risk analysis.

7.2 NFC interface guidelines

This clause contains design guidelines that apply to NFC physical devices. These can be Personal Health Devices implementing service components or Personal Health Gateways implementing client components.

7.2.1 NFC PHD to PHG linkage

This clause contains a guideline for NFC-IF service components to limit connections to one client component. The guideline is covered by Table 7-1.

Table 7-1 – NFC PHD to PHG linkage

Name	Description	Comments
NFC-Device-PHG-Linkage	A Continua NFC-IF service component **shall** connect with only one Continua NFC-IF client component at any given time.	The Continua reference topology as described in [H.810] restricts communication to a single client component.

7.2.2 NFC User experience

NFC PHDs and PHGs communicate in close proximity which is normally caused by the user bringing a NFC-IF service component PHD close to a NFC-IF client component PHG, or vice versa. This clause contains design guidelines that strongly recommend specific device behaviour to ensure a satisfying user experience. The guidelines are covered by Table 7-2.

Table 7-2 – NFC User experience

Name	Description	Comments
NFC-Device-Taptime	A Continua NFC-IF service component **should** complete data exchange within 3 seconds	Completion of data exchange within an acceptable amount of time is specifically important where the user must hold NFC service and client components in proximity for the duration of the data exchange
NFC-User-Notification	Continua NFC-IF service and client components with appropriate UI capabilities **should** notify the user when data exchange is completed	Appropriate user notifications are specifically important where the user must hold NFC service and client components in proximity for the duration of the data exchange

7.2.3 NFC personal health device communication

This clause contains a general design guideline that points to [NFC PHDC]. All subsequent requirements in Clause 7.2.3 refer to this specification. The guideline is covered in Table 7-3.

Table 7-3 – NFC personal health device communication map

Name	Description	Comments
NFC-PHDC-Map	Continua NFC-IF wireless service and client components **shall** implement NFC personal health device communication version 1.0 subject to the design guidelines below	

7.2.4 NFC Multi-function devices

This clause defines how devices that implement more than one IEEE 11073 PHD device specialization are represented via [NFC PHDC]. These guidelines require that all multi-function devices expose all device specializations via a single [IEEE 11073-20601] association. In NFC, a single [IEEE 11073-20601] association maps best to a single NFC PHDC agent interface. Thus, a Continua-certified NFC PHDC device has only one NFC PHDC agent interface for Continua functionality, regardless of whether it exposes a single device specialization or multiple device specializations. The guideline is covered in Table 7-4.

Table 7-4 – NFC Multi-function devices

Name	Description	Comments
NFC-11073-20601-Multi-Function	A Continua NFC-IF service component **shall** have at most one [IEEE 11073-20601] association to a NFC-IF client component at any point in time regardless of whether the device is a single function or multi-function device	This guideline prohibits the device from having two concurrent associations. The device may provide different configuration options only in subsequent associations only after closing the currently active association

7.2.5 NFC Quality of service

The requirements in Table 7-5 describe how Quality of Service (QoS) attributes are used for Continua NFC-IF service and client components.

Table 7-5 – NFC Quality of service

Name	Description	Comments
NFC-PHDC-QoS-Best.Medium	Continua NFC-IF service and client components **shall** provide the Continua best.medium QoS bin	NFC PHDC transport does exchange all data on best.medium QoS bin
NFC-PHDC-QoS-Good.Medium	Continua NFC-IF service and client components **shall not** provide the Continua good.medium QoS bin	NFC PHDC transport does exchange all data on best.medium QoS bin

7.3 NFC Certified Capability Classes

Table 7-6 shows the certified Capability Classes defined for the NFC interface design guidelines. A certification program run by Continua Health Alliance exists for devices that implement the CDG. For NFC devices, the certification testing will be performed on an integrated device, meaning the testing and certification is applied to the hardware and software of the device. Changes to components of the device may require a re-certification. Table 7-6 also references the guidelines that are applicable for each of the certified Capability Classes. An empty table entry would indicate that there is currently no certified Capability Class defined.

Table 7-6 – NFC Certified Capability Classes

Certified Capability Classes	Relevant guidelines
NFC activity hub service NFC activity hub client	6.2, 6.3.14, 7.2
NFC adherence monitor service NFC adherence monitor client	6.2, 6.3.29, 7.2
NFC basic 1-3 lead ECG Service NFC basic 1-3 lead ECG client	6.2, 6.3.2, 7.2
NFC blood pressure monitor service NFC blood pressure monitor client	6.2,, 6.3.4, 7.2
NFC cardiovascular fitness service NFC cardiovascular fitness client	6.2, 6.3.11, 7.2
NFC cardiovascular fitness step counter service NFC cardiovascular fitness step counter client	6.2, 6.3.12, 7.2
NFC CO sensor service NFC CO sensor client	6.2, 6.3.27, 7.2
NFC contact closure sensor service NFC contact closure sensor client	6.2, 6.3.18, 7.2
NFC continuous glucose monitor service NFC continuous glucose monitor client	6.2, 6.3.31, 7.2
NFC enuresis sensor service NFC enuresis sensor client	6.2, 6.3.17, 7.2
NFC fall sensor service NFC fall sensor client	6.2, 6.3.15, 7.2
NFC gas sensor service NFC gas sensor client	6.2, 6.3.28, 7.2
NFC glucose meter service NFC glucose meter client	6.2, 6.3.7, 7.2
NFC heart-rate sensor service NFC heart-rate sensor client	6.2, 6.3.3, 7.2
NFC INR meter service NFC INR meter client	6.2, 6.3.8, 7.2
NFC insulin pump service NFC insulin pump client	6.2, 6.3.32, 7.2

Certified Capability Classes	Relevant guidelines
NFC medication dosage sensor service NFC medication dosage sensor client	6.2, 6.3.20, 7.2
NFC motion sensor service NFC motion sensor client	6.2, 6.3.16, 7.2
NFC peak flow meter service NFC peak flow meter client	6.2, 6.3.10, 7.2
NFC PERS sensor service NFC PERS sensor client	6.2, 6.3.26, 7.2
NFC property exit sensor service NFC property exit sensor client	6.2, 6.3.23, 7.2
NFC pulse oximeter service NFC pulse oximeter client	6.2, 6.3.1, 7.2
NFC smoke sensor service NFC smoke sensor client	6.2, 6.3.22, 7.2
NFC strength fitness service NFC strength fitness client	6.2, 6.3.13, 7.2
NFC switch sensor service NFC switch sensor client	6.2, 6.3.19, 7.2
NFC temperature sensor service NFC temperature sensor client	6.2, 6.3.24, 7.2
NFC thermometer service NFC thermometer client	6.2, 6.3.5, 7.2
NFC usage sensor service NFC usage sensor client	6.2, 6.3.25, 7.2
NFC water sensor service NFC water sensor client	6.2, 6.3.21, 7.2
NFC weighing-scales service NFC weighing-scales client	6.2, 6.3.6, 7.2

8 USB interface design guidelines

8.1 USB interface architecture (informative)

This clause lists the design guidelines specific for interoperability between certified PHDs and PHGs when using USB across the Personal Health Devices Interface.

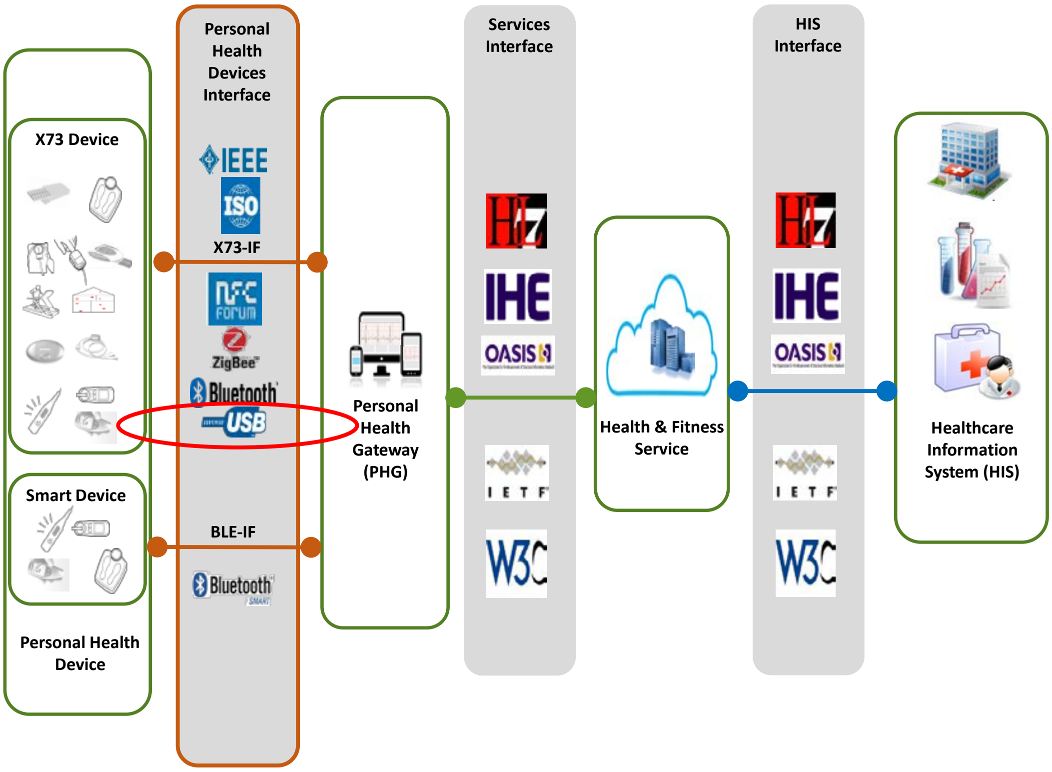

Figure 8-1 – USB interface context

8.1.1 Overview

The connectivity in the USB interface (USB-IF) is tailored to satisfying three basic requirements that are uniform across the application domains serviced by CDG-certified products:

– allow bidirectional sensor control
– allow bidirectional sensor information exchange
– allow appropriate linkage between a Personal Health Device and an Personal Health Gateway

The interface is further structured into three distinct layers, with appropriate standards selected to represent the individual layers and establish interoperability in the personal health ecosystem. Figure 8-1 illustrates the structure of the USB interface.

8.1.2 Exchange protocols and selected standards

For the data and messaging layer of the standard USB-IF, the standards from the IEEE 11073 Personal Health Device family of standards have been selected. For the detailed list of selected data/messaging layer standards, see Clause 6.

8.1.3 USB Device communication styles

The protocols selected in the USB-IF permits the device to transfer data in the following three communication styles:

- Transaction communication style: When it is required that the transport between the PHD and the PHG communicates a single data point immediately.
- Streaming communication style: When it is required that the transport between the PHD and the PHG communicates several data points continuously.
- Batch communication style: When it is required that the transport between the PHD and the PHG communicates previously collected data points at a later time.

The specific requirements pertaining to the QoS for USB for the various communication styles is outlined in Clause 8.2.5.

8.1.4 USB-IF security

For a USB solution, it is assumed that the physical action of the user connecting a USB PHD to the PHG provides the necessary security to prevent inadvertent leakage of data to a different PHG.

8.2 USB Device and interface guidelines

This clause contains design guidelines that apply to USB physical devices. These can be Personal Healthcare Devices or Personal Health Gateways.

8.2.1 USB Device to PHG linkage

Table 8-1 – USB Device to PHG linkage

Name	Description	Comments
USB-Device-PHG-Linkage	A Continua USB-IF service component **shall** connect with only one Continua USB-IF client component at any given time.	The Continua reference topology as described in [H.810] restricts communication to a single client component.

8.2.2 USB general requirements

This clause contains a general design guideline that points to the USB personal healthcare Capability Class (PHDC) v1.0 [USB DevClass]. All subsequent requirements in Clause 8.2 refer to this specification.
For more information about [USB DevClass] device drivers please see Appendix III and in [b-CHA USB-PHDC].

Table 8-2 – USB personal healthcare Capability Class v1.0 map

Name	Description	Comments
USB-Personal Healthcare-v1.0	Continua USB-IF service and client components **shall** implement the USB personal healthcare Capability Class v1.0 plus the Feb. 15, 2008 errata, subject to the requirements listed below	

8.2.3 USB Map to ISO/IEEE 11073-20601

This clause requires that a Continua-compliant device send only [IEEE 11073-20601] data and messages over USB PHDC. In addition, driver software implementing the USB PHDC transport should not need to parse the [IEEE 11073-20601] data to fully function.

Table 8-3 – ISO/IEEE 11073-20601 messaging layer

Name	Description	Comments
USB-PHDC-20601-Map-Service	Continua USB-IF service components **shall** set the USB PHDC v1.0 bPHDCDataCode field of the PHDC Class Function descriptor equal to PHDC_11073_20601	
USB-PHDC-20601-Map-Client	Continua USB-IF client components **shall** accept PHDC Class Function descriptors with the USB PHDC v1.0 bPHDCDataCode field equal to PHDC_11073_20601	
USB-PHDC-20601-Device-Spec-Cert-Dev-Classes	Continua USB-IF service components **shall** set the wDevSpecializations field(s) to the corresponding [IEEE 11073-20601] *MDC_DEV_SPEC_PROFILE_** value(s) corresponding to the certified Capability Class(es) that the component supports	
USB-PHDC-20601-Device-Spec-Not-Cert	Continua USB-IF service components **may** add additional [IEEE 11073-20601] *MDC_DEV_SPEC_PROFILE_** value(s) corresponding to supported IEEE specializations that are not Continua certified in the wDevSpecializations array	
USB-PHDC-20601-10101-Client	Continua USB-IF client components **shall not** pre-filter and reject a service component based on the wDevSpecializations field(s) value(s)	The rejection of unsupported device specializations happens in the higher layers via the [IEEE 11073-20601] Optimized exchange protocol
USB-EndOfTransfer	Continua USB-IF service and client components **shall** signify the end of a bulk transfer by transferring a payload of size less than wMaxPacketSize or a zero-length packet	USB-IF service and client components are not required to read the [IEEE 11073-20601] data to obtain the length

8.2.4 Sending metadata via USB PHDC

The USB PHDC specification [USB DevClass] contains a feature to enable the sending of QoS information with IEEE 11073 [IEEE 11073-20601] data and messages. The USB PHDC specification states that this feature is optional for service components to support and mandatory for client components to support.

It is not expected that Continua USB-IF service components will implement the feature or Continua USB-IF client components will enable the feature; however, if a service component or client component chooses to make use of the feature, the following design guidelines apply.

Table 8-4 – Using USB PHDC metadata/QoS feature

Name	Description	Comments
USB-PHDC-Enable-Meta-Data-Preamble	Continua USB-IF client components that choose to enable the USB PHDC Meta-Data Message Preamble feature **shall** attempt to enable the feature by sending the USB PHDC SET_FEATURE (FEATURE_PHDC_METADATA) request after the [IEEE 11073-20601] Association Request message has been received and before it sends the [IEEE 11073-20601] Association Response message	
USB-PHDC-Disable-Meta-Data-Preamble	Continua USB-IF client components that choose to enable the USB PHDC Meta-Data Message Preamble feature **shall** disable the feature *when in the Unassociated state* only by sending the USB PHDC CLEAR_FEATURE (FEATURE_PHDC_METADATA) request	
USB-bQoSEncodingVersionOOB	Continua USB-IF client components that receive a bQoSEncodingVersion field that is not 01h **shall** ignore the bmLatencyReliability bitmap as it could have a different meaning in a future version of the specification	This replaces the text "In order to remain forward compatible, if a host implementing 01h QoS information encoding receives a bQoSEncodingVersion field that is not 01h, it shall ignore the descriptor." on page 22, 1st paragraph, of [USB DevClass]

8.2.5 USB Quality of service

The following requirements describe how QoS attributes are used for Continua USB-IF service and client components.

Table 8-5 – Mapping of USB PHDC QoS bins into Continua QoS bins

Name	Description	Comments
USB-QoS-Best.Medium	Continua USB-IF service and client components that implement the Continua *best.medium* QoS bin **shall** utilize the USB PHDC *best.medium* QoS bin to do this	
USB-QoS-Good.Medium	Continua USB-IF service and client components that implement the Continua *good.medium* QoS bin **shall** utilize the USB PHDC *good.medium* QoS bin to do this	

8.2.6 USB Multi-function devices

This clause defines how devices that implement more than one IEEE 11073 PHD device specialization are represented via USB PHDC. CDG requires that all multi-function devices expose all device specializations via a single [IEEE 11073-20601] association. In USB, a single [IEEE 11073-20601] association maps best to a single USB PHDC interface. Thus, a Continua-certified USB PHDC device has only one USB PHDC interface for CDG functionality, regardless of whether it exposes a single device specialization or multiple device specializations. This is shown in Figure 8-2.

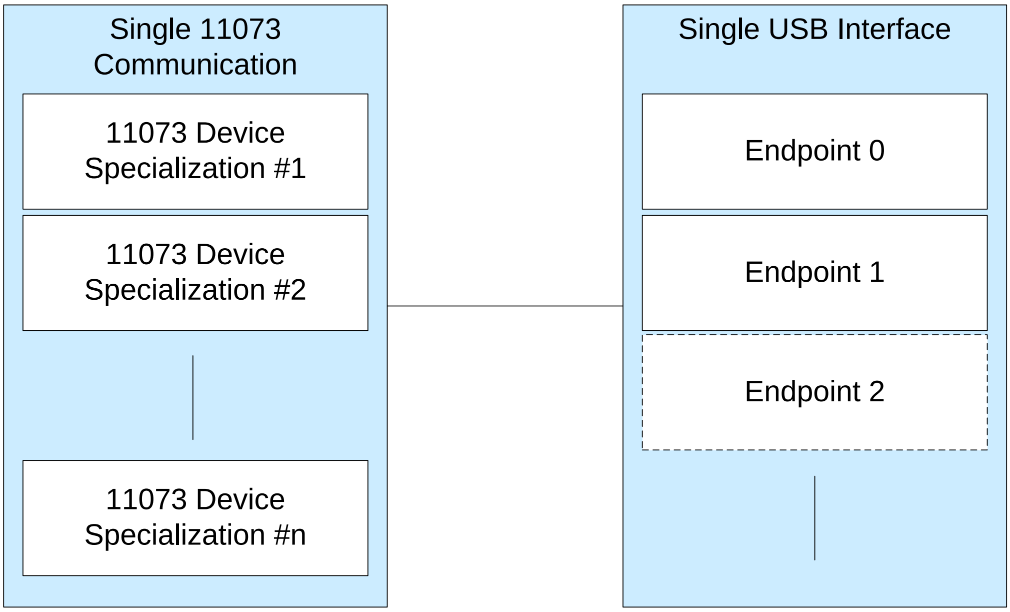

Figure 8-2 – USB PHDC mapping to [IEEE 11073-20601] associations

Table 8-6 – USB Multi-function devices

Name	Description	Comments
USB-PHDC-Multi-Function-Single-Interface	Continua USB-IF service components, whether multi-function or single function, **shall** implement one and only one USB PHDC interface for the component's [IEEE 11073-20601] association	CDG requires that all USB multi-function devices expose all functions via a single [IEEE 11073-20601] association. See 11073-20601-Multi-Function.

8.2.7 USB Connectors

USB contains a few connector options on the service and client side. The following design guidelines give guidance on connector choices for implementation.

Table 8-7 – USB connectors

Name	Description	Comments
USB-B-Connector-Connectivity	A Continua USB PHD **should** be shipped with a mechanism for connecting themselves to an PHG assuming a standard-A connector to the PHG	Example connectivity mechanisms include a cable that connects to the device and exposes a standard-A connector and an integral cable on the device that exposes a standard-A connector
USB-B-Connector-Mechanism-to-Obtain-Connectivity	If a Continua USB PHD does not ship with a mechanism for connectivity as defined in USB-B-Connector-Connectivity, it **shall** ship with a mechanism for obtaining such connectivity	Example mechanisms for obtaining connectivity include documentation on the type of cable needed and possibly, a phone number, mail in the form or website for requesting and/or purchasing that cable
USB-A-Connector-Connectivity	Continua USB PHGs that do not accept a Standard-A female connector **should** be shipped with a mechanism for converting to accept a Standard-A female connector	Example mechanisms include a converter from the A connector on the PHG to standard-A
USB-A-Connector-Mechanism-to-Obtain-Connectivity	If a Continua USB PHG that does not accept a Standard-A female connector does not ship with a mechanism for converting to Standard-A female connector, it **shall** be shipped with a mechanism for obtaining a conversion to accept a Standard-A female connector	Example mechanisms include documentation on the converter necessary, and possibly, a phone number, mail in the form or website for requesting and/or purchasing that converter

8.2.8 USB Data rates

USB 2.0 provides full speed and high speed data rates. USB 1.1 provides low speed and full speed data rates. This clause describes the requirements CDG places on the data rates to use.

Table 8-8 – USB data rates

Name	Description	Comments
USB-Low-Speed	Continua USB-IF service and client components **shall not** use low speed	Low speed is mostly used for keyboards, mice, and joysticks. Low speed does not support all data rates required by the CDG. Max packet size for low-speed is 8 bytes. Low-speed also has behavioural differences with full and high speed. NOTE - Low speed is only available in USB 1.1
USB-USB-2.0	Continua USB-IF service and client components **should** implement USB 2.0	
USB-USB-1.1	Continua USB-IF service and client components **shall** implement at least USB 1.1 or any superior version compatible with USB 1.1	

8.3 USB Certified Capability Classes

Table 8-9 shows the certified Capability Classes defined for the USB-IF design guidelines. A certification program run by Continua Health Alliance exists for devices that implement the CDG. For USB PHDs and PHGs, the certification testing will be performed on an integrated device, meaning the testing and certification is applied to the hardware and software of the device. Changes to components of the device may require a re-certification. Table 8-9 also references the guidelines that are applicable for each of the certified Capability Classes.

Table 8-9 – USB Certified Capability Classes

Certified Capability Classes	USB *(relevant guidelines)*
USB activity hub service USB activity hub client	6.2, 6.3.14, 8.2
USB adherence monitor service USB adherence monitor client	6.2, 6.3.29, 8.2
USB basic 1-3 lead ECG service USB basic 1-3 lead ECG client	6.2, 6.3.2, 8.2
USB blood pressure monitor service USB blood pressure monitor client	6.2, 6.3.4, 8.2
USB cardiovascular fitness service USB cardiovascular fitness client	6.2, 6.3.11, 8.2
USB cardiovascular fitness step counter service USB cardiovascular fitness step counter client	6.2, 6.3.12, 8.2
USB CO sensor service USB CO sensor client	6.2, 6.3.27, 8.2

Certified Capability Classes	USB *(relevant guidelines)*
USB contact closure sensor service USB contact closure sensor client	6.2, 6.3.18, 8.2
USB continuous glucose monitor service USB continuous glucose monitor client	6.2, 6.3.31, 8.2
USB enuresis sensor service USB enuresis sensor client	6.2, 6.3.17, 8.2
USB fall sensor service USB fall sensor client	6.2, 6.3.15, 8.2
USB gas sensor service USB gas sensor client	6.2, 6.3.28, 8.2
USB glucose meter service USB glucose meter client	6.2, 6.3.7, 8.2
USB heart-rate sensor service USB heart-rate sensor client	6.2, 6.3.3, 8.2
USB INR meter service USB INR meter client	6.2, 6.3.8, 8.2
USB insulin pump service USB insulin pump client	6.2, 6.3.32, 8.2
USB medication dosage sensor service USB medication dosage sensor client	6.2, 6.3.20, 8.2
USB motion sensor service USB motion sensor client	6.2, 6.3.16, 8.2
USB peak flow meter service USB peak flow meter client	6.2, 6.3.10, 8.2
USB PERS sensor service USB PERS sensor client	6.2, 6.3.26, 8.2
USB property exit sensor service USB property exit sensor client	6.2, 6.3.23, 8.2
USB pulse oximeter service USB pulse oximeter client	6.2, 6.3.1, 8.2
USB SABTE service USB SABTE client	6.2, 6.3.30, 8.2
USB smoke sensor service USB smoke sensor client	6.2, 6.3.22, 8.2
USB strength fitness service USB strength fitness client	6.2, 6.3.13, 8.2
USB switch sensor service USB switch sensor client	6.2, 6.3.19, 8.2
USB temperature sensor service USB temperature sensor client	6.2, 6.3.24, 8.2

Certified Capability Classes	USB *(relevant guidelines)*
USB thermometer service USB thermometer client	6.2, 6.3.5, 8.2
USB usage sensor service USB usage sensor client	6.2, 6.3.25, 8.2
USB water sensor service USB water sensor client	6.2, 6.3.21, 8.2
USB weighing-scales service USB weighing-scales client	6.2, 6.3.6, 8.2

9 Bluetooth BR/EDR interface design guidelines

9.1 Bluetooth BR/EDR interface architecture (informative)

This clause lists the design guidelines specific for interoperability between certified PHDs and PHGs when using Bleutooth BR/EDR across the Personal Health Devices Interface.

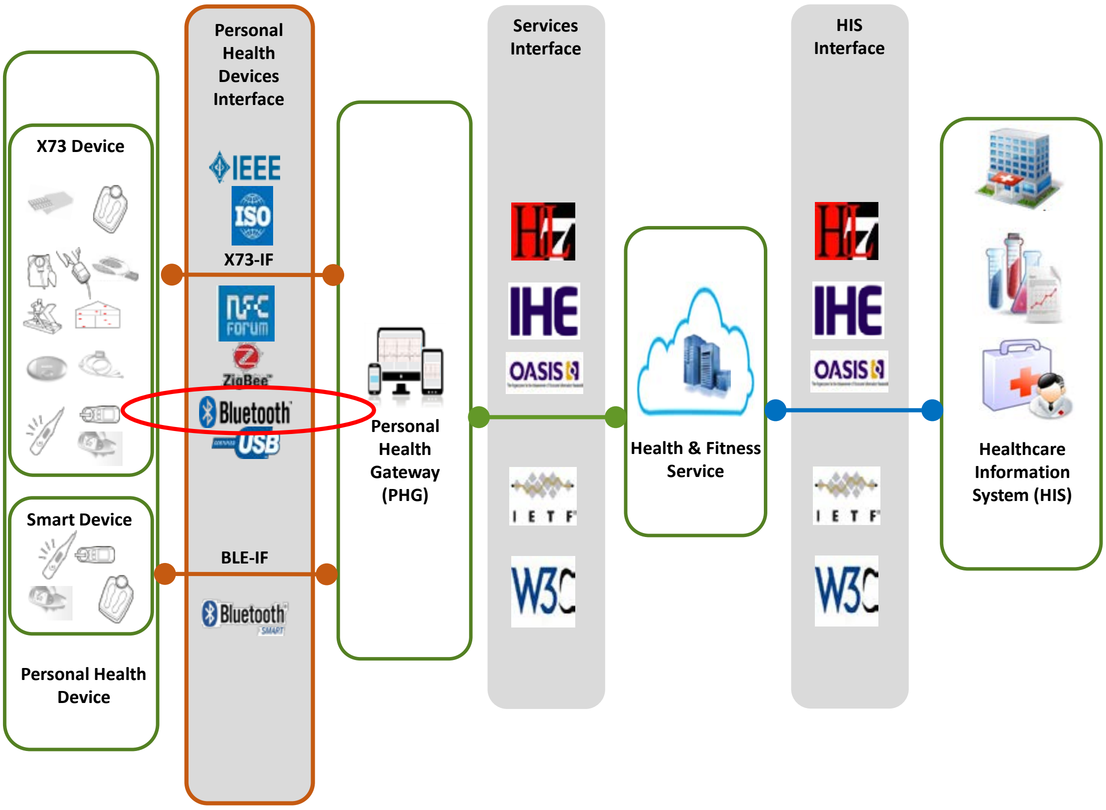

Figure 9-1 – Bluetooth interface

9.1.1 Overview

The connectivity in the Bluetooth BR/EDR interface (BR/EDR-IF) is tailored to satisfying three basic requirements that are uniform across the application domains serviced by CDG-certified products:

– allow bidirectional sensor control
– allow bidirectional sensor information exchange
– allow appropriate linkage between a Personal Health Device and an Personal Health Gateway

The interface is further structured into three distinct layers, with appropriate standards selected to represent the individual layers and establish interoperability in the personal health ecosystem. Figure 9-1 illustrates the structure of the Bluetooth interface.

9.2 Bluetooth BR/EDR interface guidelines

9.2.1 Bluetooth BR/EDR PHD to PHG linkage

Table 9-1 – Bluetooth BR/EDR PHD to PHG linkage

Name	Description	Comments
ContinuaStructType	A Continua BR/EDR-IF service component **shall** connect with only one Continua BR/EDR-IF client component at any given time.	**9.2.2** The Continua reference topology as described in [H.810] restricts communication to a single client component.

9.2.3 Bluetooth Health Device Profile

This clause contains a general design guideline that points to [Bluetooth HDPv1.1. All subsequent requirements in Clause 9.2 refer to this specification. For further guidance on implementing the Bluetooth Health Device Profile the reader is referred to the white paper [b_Bluetooth HDPIP].
Throughout this clause, some common Bluetooth terms are used:
When the term "discovery" is used, this is meant to describe its use of the Bluetooth inquiry substate to learn of the existence of other Bluetooth devices within transmission range. This is sometimes called "device discovery" to distinguish from service discovery. A Bluetooth device is discoverable if it periodically enters the inquiry Scan substate. A discoverable device will respond to inquiry procedures (usually a general inquiry) from any device that wants to search.
A Bluetooth device enters the inquiry substate to discover other Bluetooth devices. Discoverable devices will periodically enter the inquiry scan substate.
Service discovery creates a baseband connection to a specific device (may be paired, but does not need to be) to discover details about services offered on that device.
When the term "pairing" is used, this is meant to describe the exchange of link keys to establish a future trust relationship with a known device. Except in legacy cases, this is performed with secure simple pairing (SSP).
When the term "connectable" is used, this is meant to describe a previously paired device that is periodically entering the page scan substate and responds to pages from devices that address it specifically (by Bluetooth MAC address). For a device to be connected, it must first be paired.

Table 9-2 – Bluetooth health device profile map

Name	Description	Comments
Bluetooth-BR/EDR-Map	Continua BR/EDR-IF service and client components **shall** be compliant with Bluetooth 2.1.	Later versions of the Bluetooth specification can be used as long as version 2.1 functionality is fully supported.
Bluetooth-BR/EDR-HDP-Map	Continua BR/EDR-IF service and client components **shall** be compliant with Bluetooth Health Device Profile version 1.1 subject to the design guidelines below.	Later versions of the Bluetooth HDP specification can be used as long as version 1.1 functionality is fully supported.

9.2.4 Discovery and pairing

Continua X73 Bluetooth BR/EDR devices transfer measurement data to partner devices. These partnerships are formed either following a search initiated by the client component that will receive the data or through an out-of-band configuration.

This guidelines document requires a process of discovery of the service component by the client component for all Bluetooth CDG devices. This ensures a consistent and user-friendly pairing procedure.

The guidelines throughout this clause create a single and universally supported technique for pairing devices that give a minimum of surprise or inconvenience to users. These guidelines apply to Bluetooth versions 2.0 and 2.1.

Table 9-3 – Bluetooth BR/EDR pairing guidelines

Name	Description	Comments
Bluetooth-BR/EDR-Discovery-Initiation-Client	Continua BR/EDR-IF client components **shall** initiate discovery (a Bluetooth “Inquiry”)	
Bluetooth-BR/EDR-Discovery-Initiation-Service	Continua BR/EDR-IF service components **should not** initiate discovery (a Bluetooth "Inquiry")	
Bluetooth-BR/EDR-Pairing-Service	Continua BR/EDR-IF service components **shall** have a documented way (decided by the vendor) to initiate a mode of "discoverable by the client component" Once a service component has been made discoverable in this way, it **shall** support pairing with compatible client components, as shown in Figure 9-2	The words 'compatible client components' refer to client components that share the same device specialization as the service component
Bluetooth-BR/EDR-Pairing-Client	Continua BR/EDR-IF client components **shall** have a documented way (decided by the vendor) to initiate a search for service components that are "discoverable" Once the client component has discovered such service component, it **shall** support pairing with compatible service components, as shown in Figure 9-3	The words 'compatible service components' refer to service components that share the same device specialization as the client component Client components may be pre-configured to pair with a specific service component; however, they are required to provide support for discovery and pairing of any compatible service component.

Name	Description	Comments
Bluetooth-BR/EDR-All-Pairing-Client	Continua BR/EDR-IF client components **shall** support all pairing methods for Bluetooth 2.1, including Just Works, Numeric Comparison, and Passkey Entry, if the client component has the appropriate I/O capabilities	I/O capabilities include display, keyboard, yes/no. See the Bluetooth core specification [Bluetooth CS2.1] and secure simple pairing white papers for further information. This pairing guideline is necessary to ensure interoperability and give reasonable assurance that a service component's chosen pairing method will be supported by client components
Bluetooth-BR/EDR-Legacy-Pairing-Client	Continua BR/EDR-IF client components **shall** support legacy (Bluetooth 2.0) pin entry pairing	This guideline is necessary to ensure backward compatibility with existing Continua BT 2.0 service components
Bluetooth-BR/EDR-Pairing-Service-2	Continua BR/EDR-IF service components **shall** support at least one of the following Bluetooth 2.1 pairing methods depending on their I/O capabilities and appropriate security for the service component device type: Just Works, Numeric Comparison, or Passkey Entry	I/O capabilities include display, keyboard, yes/no. See the Bluetooth core specification [Bluetooth CS2.1] and secure simple pairing white papers for further information
Bluetooth-BR/EDR-Re-Pairing	Once a Continua BR/EDR-IF service component has been paired with a client component, it **shall** remain possible to re-initiate the mode "discoverable by the client component"	
Bluetooth-BR/EDR-Data-Exchange-Service	Continua BR/EDR-IF service component data (not including HDP service discovery record or static information like capabilities, service names, etc.) **shall not** be exchanged with client components for which a pairing has not been established	
Bluetooth-BR/EDR-Discoverability-Mode-Service	By default, Continua BR/EDR-IF service components **should not** be discoverable unless put in that mode as documented above	
Bluetooth-BR/EDR-Discoverability-Mode-Client	Continua BR/EDR-IF client components **should not** be discoverable unless put in that mode as documented above	

Name	Description	Comments
Bluetooth-BR/EDR-Discoverability-Duration	Continua BR/EDR-IF service components **should** provide a documented minimum duration (decided by the vendor) for this discoverable mode, once initiated, after which it ceases to be discoverable	
Bluetooth-BR/EDR-Paired	When a Continua BR/EDR-IF service component is discoverable and successfully completes a pairing procedure, it **should** immediately become undiscoverable	

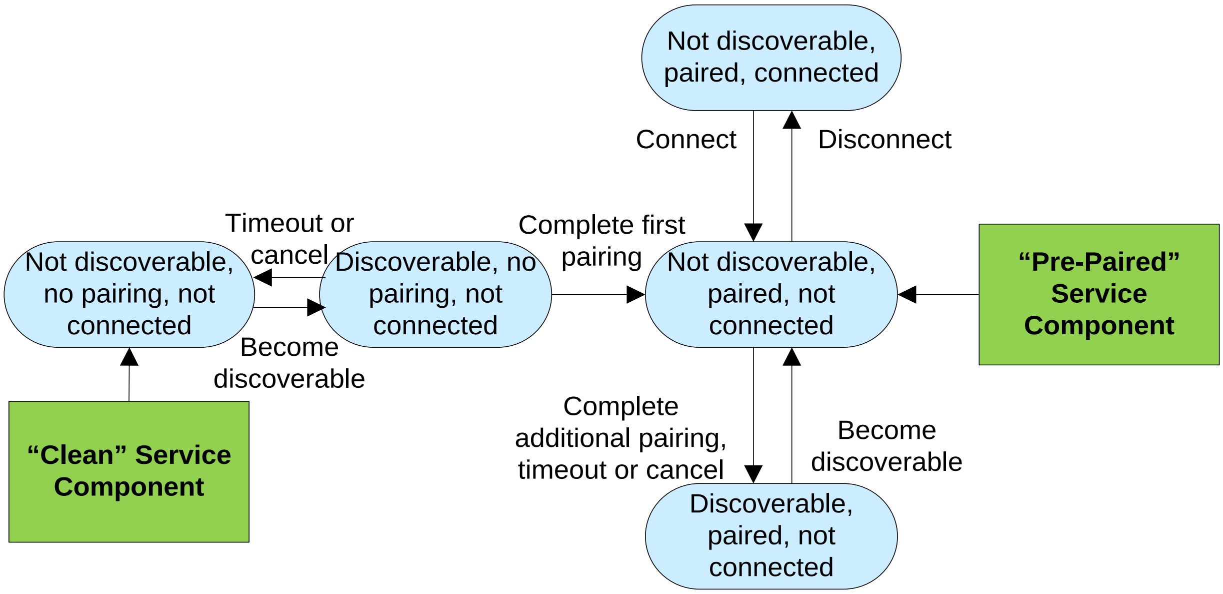

Figure 9-2 – Continua Bluetooth BR/EDR pairing process for service components

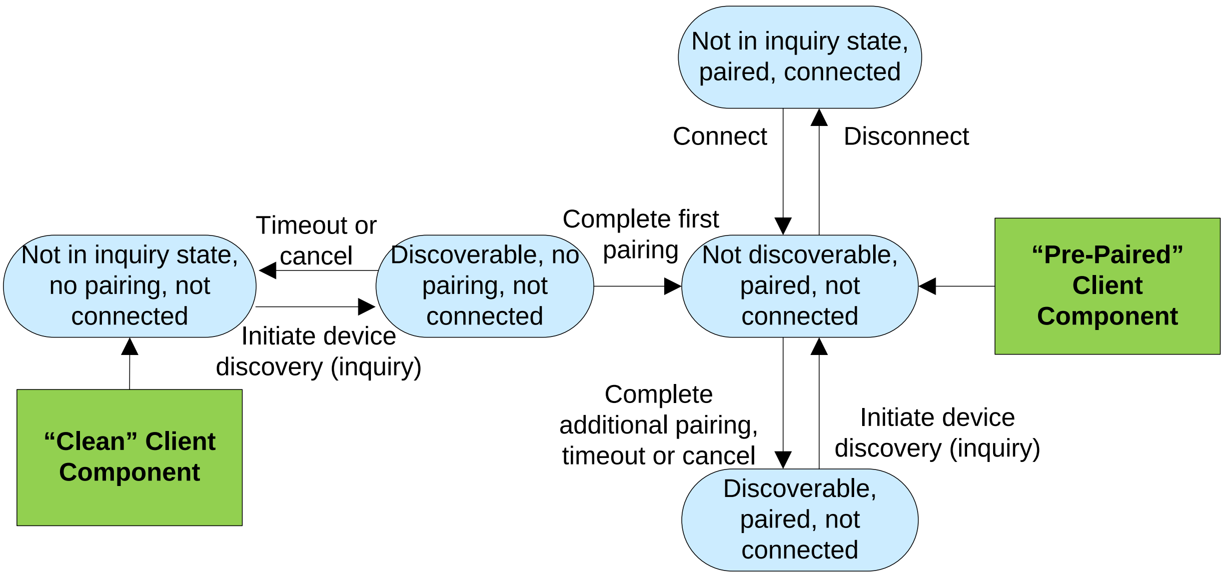

Figure 9-3 – Continua Bluetooth BR/EDR pairing process for client components

The diagram in Figure 9-2 shows the behaviour of a Continua BR/EDR-IF service component in the pairing process and the diagram in Figure 9-3 shows the behaviour of a BR/EDR-IF client component in the pairing process. Some Bluetooth BR/EDR devices may permit pairing from non-discoverable states, if the partner device knows the MAC address of the service component (either through out-of-band configuration or from a previous device discovery operation). These transitions are not shown, although technically possible, for simplicity. Because they represent a non-standard operation of the device, they may present a security vulnerability for some applications.

Table 9-4 – Bluetooth BR/EDR pairing in non-discoverable states

Name	Description	Comments
Bluetooth-BR/EDR-Non-Discovery	If a Continua BR/EDR-IF service component is able to prevent pairing while in non-discoverable states, it **should** do so	

The reason for this procedure is to provide security and privacy for users while optimizing the ease of use by providing predictable behaviour, and by minimizing the time and effort required to execute the pairing.

Another ease-of-use issue is the frequency required for a user to go through the pairing procedure. To avoid unnecessary re-pairings following battery replacements or power failures, persistent storage on sensors is important.

Table 9-5 – Bluetooth BR/EDR pairing data

Name	Description	Comments
Bluetooth-BR/EDR-Pairing-Data-Service	Continua BR/EDR-IF service components **shall** store the pairing data from at least the most recently paired device in such a way that the data will be retained through normal power interruptions, including battery replacement	
Bluetooth-BR/EDR-Pairing-Data-Client	Continua BR/EDR-IF client components **shall** store the pairing data from at least the most recently paired device in such a way that the data will be retained through normal power interruptions, including battery replacement Continua BR/EDR-IF client components **should** store pairing data for at least the number of devices for which they are intended to simultaneously support	

9.2.5 Bluetooth BR/EDR discoverable mode

The requirements in the previous clause refer to a mode where a device is "discoverable by the client component." In Bluetooth terms, this means the device is in both "discoverable mode" and "pairable mode" (also known as "bondable mode"). When a device is in *Bluetooth* "discoverable mode," other devices can perform inquiries to learn its MAC address. From a CDG point of view, since all communication is between paired devices, it does not make sense for a service component to be discoverable unless it is willing to pair with devices that discover it.

Leaving a device in the discoverable (and pairable) state opens the device to hackers who may attempt to connect. Being discoverable is a security risk, as well as a privacy risk.

Table 9-6 – Bluetooth BR/EDR discovery disable

Name	Description	Comments
Bluetooth-BR/EDR-Discovery-Disable	Continua BR/EDR-IF service components that may become discoverable in the course of normal use **should** offer users a mechanism to disable this behaviour	

To avoid pairing with devices that cannot be used, it is helpful for devices to allow access to their HDP service discovery protocol (SDP) record to enable a connecting device, to query the capability of devices and identify the device specializations supported.

Table 9-7 – Bluetooth SDP access

Name	Description	Comments
Bluetooth-BR/EDR-SDP-Access	When possible, Continua BR/EDR-IF service components in "discoverable mode" **should** allow access to their SDP entries without first requiring a pairing to be established	

The Bluetooth HDP SDP record includes a list of supported [ISO/IEEE 11073-104xx] specializations under the SDP attribute "MDEP Data Type". This list is used to filter devices for suitability and is required by the Bluetooth HDP specification [Bluetooth HDPv1.1] to match the list of [ISO/IEEE 11073-104xx] specializations actually supported by the implementation.

Table 9-8 – Bluetooth SDP record

Name	Description	Comments
Bluetooth-BR/EDR-SDP-Record	The specializations claimed in Continua certification **shall** match the list of specializations advertised in the Continua BR/EDR-IF service component HDP SDP record	
Bluetooth-BR/EDR-SDP-Extensions	The Continua BR/EDR-IF service component HDP SDP record **may** contain additional specialization identifiers that are not Continua certified	

9.2.6 Notifying the user

Establishing a new pairing relationship is an important event. Because of the potential for confusion, extreme care should be used before automating the pairing procedure. To allow users reasonable control of their Continua systems, PHGs are required to provide a facility for alerting users of significant events. Because discovery may be difficult for users to understand, it is important to inform them of new pairings and reasons for failure. The design guidelines in this clause intentionally leave the nature of notifying and informing the user to be defined by the manufacturer.

Table 9-9 – Bluetooth BR/EDR user notification

Name	Description	Comments
Bluetooth-BR/EDR-Pairing-Creation-Alert-Client	Continua BR/EDR-IF client components **shall** inform the user when a new pairing relationship is created	
Bluetooth-BR/EDR-Pairing-Creation-Alert-Service	Continua BR/EDR-IF service components **should** notify the user, whenever possible, when a new pairing relationship is created	
Bluetooth-BR/EDR-Pairing-Failure-Alert-Client	When a pairing fails, Continua BR/EDR-IF client components **shall** inform the user whether the failure was because no service component was found (discovery failed), no data types are supported in common by both the client component and service component (incompatible device), or the pairing failed (pairing failure)	
Bluetooth-BR/EDR-Pairing-Failure-Alert-Service	Whether or not pairing fails, Continua BR/EDR-IF service components **should** inform the user, whenever possible, if no data types are supported in common by both the client component and service component (incompatible device), or the pairing failed (pairing failure)	

Actual use of devices varies widely and it is not always clear which device is more physically convenient to the user during these pairing events. For this reason and also to increase the chance that a user will notice improper use of a device, pairing notifications should be made as noticeable as possible.

Table 9-10 – Bluetooth BR/EDR authentication/security failure notification

Name	Description	Comments
Bluetooth-BR/EDR-Security-Failure-Client	When any authentication/security failure is encountered by Continua BR/EDR-IF client components, client components **shall** notify the user	
Bluetooth-BR/EDR-Security-Failure-Service	When any authentication/security failure is encountered by Continua BR/EDR-IF service components, service components **should** notify the user whenever possible	

9.2.7 Quality of service

Table 9-11 – Bluetooth BR/EDR quality of service

Name	Description	Comments
Bluetooth-BR/EDR-QoS-Best.Medium	Continua BR/EDR-IF service and client components that implement the Continua *best.medium* QoS bin **shall** utilize the HDP reliable data channel type to do this	See Clause 6.1.6.2 in [H.810] for a definition of the QoS bins.
Bluetooth-BR/EDR-QoS-Good.Medium	Continua BR/EDR-IF service and client components that implement the Continua *good.medium* QoS bin **shall** utilize the HDP streaming data channel type to do this	See Clause 6.1.6.2 in [H.810] for a definition of the QoS bins

While the Bluetooth core specification [Bluetooth CS2.1] specifies the use of a 16-bit FCS by default, it is optional in HDP [Bluetooth HDPv1.1] for "Reliable" and "Streaming" data channel types to disable the FCS (frame check sequence) if both sides agree during negotiation. The baseband already uses a CRC to detect bit errors in the data frames and FCS implements a second CRC to increase the probability of error detection. While devices that can tolerate an occasional error (e.g. a pedometer counting the number of steps walked) and have limited processor or battery resources may opt not to use FCS, FCS is recommended for all other cases. This will significantly improve (estimated to be on the order of thousands of times) the probability that an error is detected.

Table 9-12 – Bluetooth BR/EDR error detection

Name	Description	Comments
Bluetooth-BR/EDR-FCS	When possible and appropriate to the device, Continua BR/EDR-IF service and client components **should** use FCS for all data channels	

9.2.8 Secure simple pairing debug mode

If a device compliant with Bluetooth version 2.1 connects to another device also compliant with Bluetooth version 2.1, the use of SSP in Bluetooth is mandatory. SSP results in an encrypted link requiring a Private Key to decrypt packets. To make the decryption of over-air packets possible for the purposes of test and debug when SSP is used (e.g. via a sniffer or protocol analyser), devices compliant with Bluetooth 2.1 would need to implement the SSP debug mode. Debug mode only needs to be supported by one of the two sides of the link for over-air decryption to be possible.

9.3 Certified Capability Classes

Table 9-13 shows the certified Capability Classes defined for the BR/EDR-IF design guidelines. A certification program run by Continua Health Alliance exists for devices that implement the CDG. For Bluetooth BR/EDR devices, the certification testing will be performed on an integrated device, meaning the testing and certification is applied to the hardware and software of the device. Changes to components of the device may require a re-certification. Table 9-13 also references the guidelines that are applicable for each of the certified Capability Classes.

Table 9-13 – Bluetooth BR/EDR Certified Capability Classes

Certified Capibility Class	relevant guidelines
Bluetooth BR/EDR activity hub service Bluetooth BR/EDR activity hub client	6.2, 6.3.14, 9.2
Bluetooth BR/EDR adherence monitor service Bluetooth BR/EDR adherence monitor client	6.2, 6.3.29, 9.2
Bluetooth BR/EDR basic 1-3 lead ECG service Bluetooth BR/EDR basic 1-3 lead ECG client	6.2, 6.3.2, 9.2
Bluetooth BR/EDR blood pressure monitor service Bluetooth BR/EDR blood pressure monitor client	6.2, 6.3.4, 9.2
Bluetooth BR/EDR cardiovascular fitness service Bluetooth BR/EDR cardiovascular fitness client	6.2, 6.3.11, 9.2
Bluetooth BR/EDR cardiovascular fitness step counter service Bluetooth BR/EDR cardiovascular fitness step counter client	6.2, 6.3.12, 9.2
Bluetooth BR/EDR CO sensor service Bluetooth BR/EDR CO sensor client	6.2, 6.3.27, 9.2
Bluetooth BR/EDR contact closure sensor service Bluetooth BR/EDR contact closure sensor client	6.2, 6.3.18, 9.2
Bluetooth BR/EDR continuous glucose monitor service Bluetooth BR/EDR continuous glucose monitor client	6.2, 6.3.31, 9.2
Bluetooth BR/EDR enuresis sensor service Bluetooth BR/EDR enuresis sensor client	6.2, 6.3.17, 9.2
Bluetooth BR/EDR fall sensor service Bluetooth BR/EDR fall sensor client	6.2, 6.3.15, 9.2
Bluetooth BR/EDR gas sensor service Bluetooth BR/EDR gas sensor client	6.2, 6.3.28, 9.2
Bluetooth BR/EDR glucose meter service Bluetooth BR/EDR glucose meter client	6.2, 6.3.7, 9.2
Bluetooth BR/EDR heart-rate sensor service Bluetooth BR/EDR heart-rate sensor client	6.2, 6.3.3, 9.2
Bluetooth BR/EDR INR meter service Bluetooth BR/EDR INR meter client	6.2, 6.3.8, 9.2
Bluetooth BR/EDR insulin pump service Bluetooth BR/EDR insulin pump client	6.2, 6.3.32, 9.2
Bluetooth BR/EDR medication dosage sensor service Bluetooth BR/EDR medication dosage sensor client	6.2, 6.3.20, 9.2
Bluetooth BR/EDR motion sensor service Bluetooth BR/EDR motion sensor client	6.2, 6.3.16, 9.2
Bluetooth BR/EDR peak flow meter service Bluetooth BR/EDR peak flow meter client	6.2, 6.3.10, 9.2
Bluetooth BR/EDR PERS sensor service Bluetooth BR/EDR PERS sensor client	6.2, 6.3.26, 9.2

Certified Capibility Class	relevant guidelines
Bluetooth BR/EDR property exit sensor service Bluetooth BR/EDR property exit sensor client	6.2, 6.3.23, 9.2
Bluetooth BR/EDR pulse oximeter service Bluetooth BR/EDR pulse oximeter client	6.2, 6.3.1, 9.2
Bluetooth BR/EDR SABTE service Bluetooth BR/EDR SABTE client	6.2, 6.3.30, 9.2
Bluetooth BR/EDR smoke sensor service Bluetooth BR/EDR smoke sensor client	6.2, 6.3.22, 9.2
Bluetooth BR/EDR strength fitness service Bluetooth BR/EDR strength fitness client	6.2, 6.3.13, 9.2
Bluetooth BR/EDR switch sensor service Bluetooth BR/EDR switch sensor client	6.2, 6.3.19, 9.2
Bluetooth BR/EDR temperature sensor service Bluetooth BR/EDR temperature sensor client	6.2, 6.3.24, 9.2
Bluetooth BR/EDR thermometer service Bluetooth BR/EDR thermometer client	6.2, 6.3.5, 9.2
Bluetooth BR/EDR usage sensor service Bluetooth BR/EDR usage sensor client	6.2, 6.3.25, 9.2
Bluetooth BR/EDR water sensor service Bluetooth BR/EDR water sensor client	6.2, 6.3.21, 9.2
Bluetooth BR/EDR weighing-scales service Bluetooth BR/EDR weighing-scales client	6.2, 6.3.6, 9.2

10 ZigBee interface design guidelines

10.1 ZigBee interface Architecture (informative)

10.1.1 Introduction

This clause lists the design guidelines specific for interoperability between Continua Certified PHDs and PHGs using the ZigBee across the Personal Health Devices Interface. Figure 10-1 illustrates the ZigBee interface in the context of the Continua E2E architecture. The ZigBee interface is a particular sub-class of the Continua PHD-IFs and connects ZigBee PHDs to PHGs across all three CDG domains, disease management, ageing independently, and health and fitness.

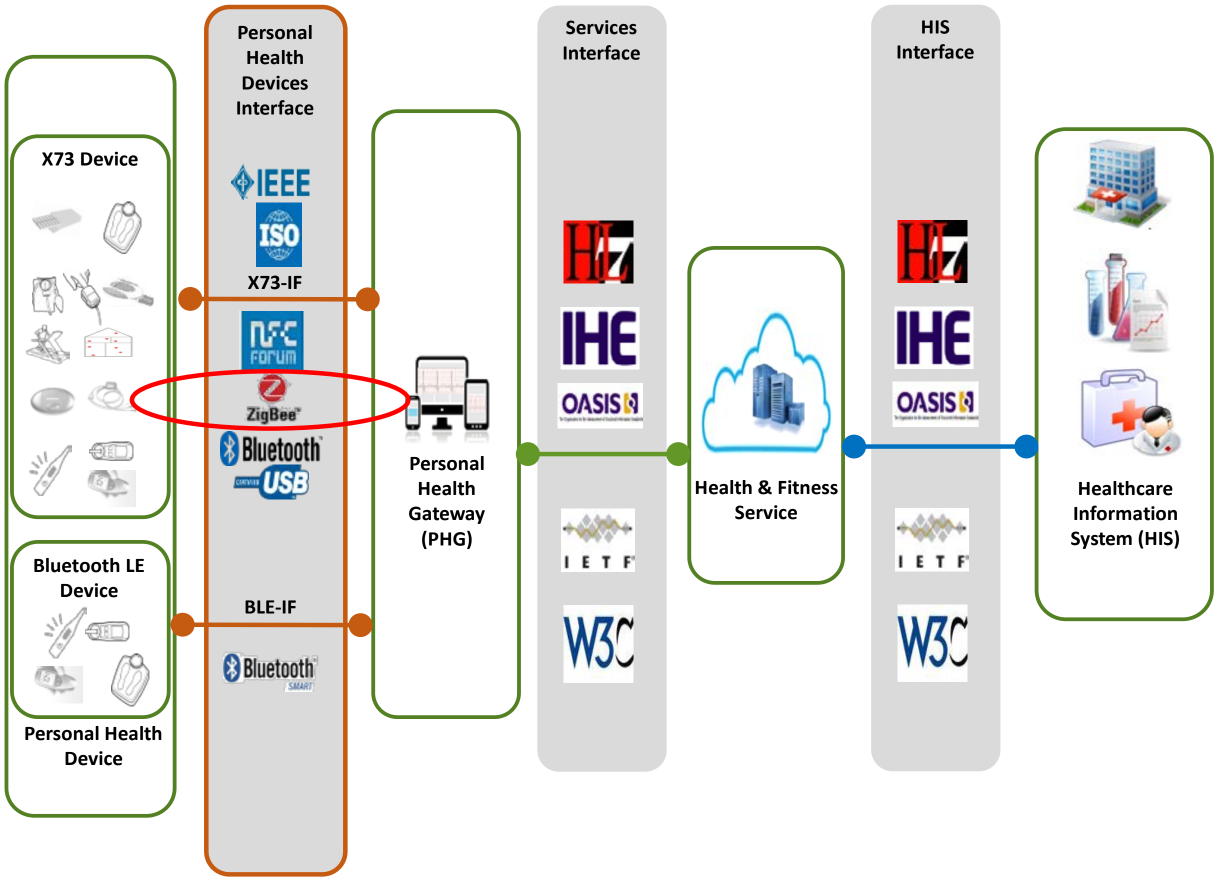

Figure 10-1 – ZigBee interface

10.1.2 Scope

The ZigBee interface enables sensors (or actuators) to send their measured data to (or to be controlled by) one or many Continua PHGs that are placed around the same house, building, facility or campus. In this respect, the ZigBee interface provides wireless infrastructure based connectivity in an area around a location. The network coverage area can scale up to several hundreds of meters, with several tens up to several thousands of devices being a part of that network. The location of sensors/actuators (PHDs) connected via the ZigBee interface can be fixed as well as mobile, with the latter case referring to devices (e.g. body worn) roaming throughout the network up to walking/running speed. Furthermore, up to years of battery lifetime is enabled for PHDs connected via the ZigBee interface. See Figure 10-2 for a high-level illustrative diagram of the ZigBee conceptual set-up. In Figure 10-2(a) ZigBee PHDs are utilizing an existing wireless infrastructure network for communication and in Figure 10-2(b) ZigBee PHDs are being part of and contributing to the wireless infrastructure network.

The use of the ZigBee interface is not limited to large-scale, long-range networks. Rather it can be used to establish direct short-range connections between PHDs and PHGs as well.

In version 2010 of the CDG, the scope of the ZigBee interface was restricted to many-to-one connectivity. According to this a PHG may connect to one or more ZigBee PHDs at the same time, but a Continua ZigBee PHD was allowed to connect to a single Continua PHG at the same time only. In this version of the CDG; the extension to many-to-many connectivity is defined, i.e., the simultaneous connection of a ZigBee PHD to multiple PHGs at the same time is supported.

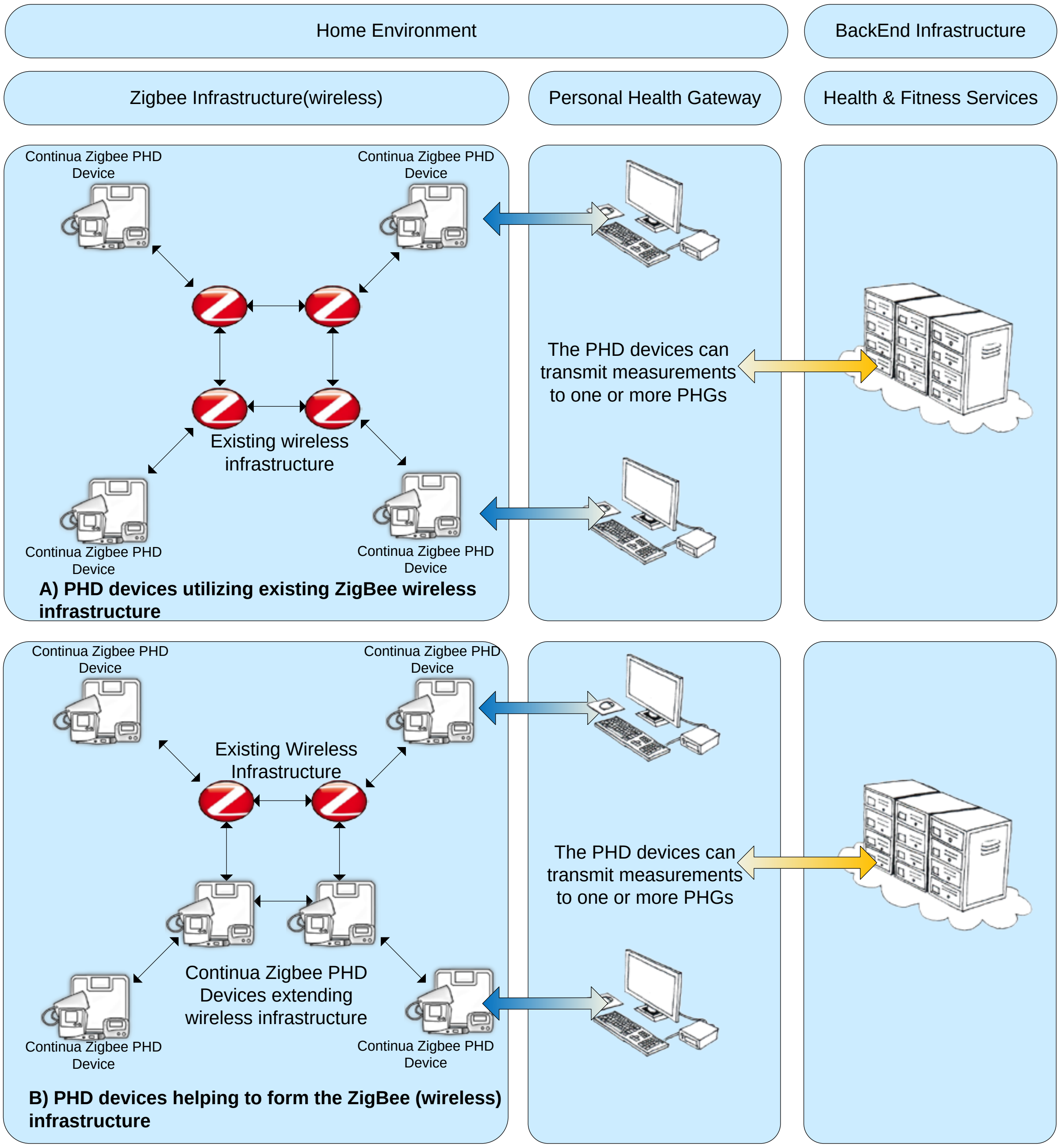

Figure 10-2 – ZigBee conceptual set-up

10.1.3 Overview

The interface is structured into distinct layers. Appropriate standards are selected for the individual layers and establish interoperability in the personal health ecosystem. See Figure 6-2 for an overview of the protocol stack of the ZigBee interface.

10.1.4 Transport protocol and selected standards

The ZigBee Health Care profile version 1.0 has been selected as the wireless lower layer protocol to serve as the transport for the ZigBee interface. The selected protocol for the transport layer ensures interoperable set-up and tear-down of the communication network for transfer of control information and transfer of data messages across all domains.

10.1.5 Data exchange protocol and selected standards

For the data and messaging layer of the ZigBee interface, the standards from the IEEE 11073 personal health device family of standards have been selected. For the detailed list of selected data/messaging layer standards please see Clause 6.

10.2 ZigBee interface guidelines

10.2.1 ZigBee transport layer

10.2.1.1 ZigBee health care profile

This clause contains a general design guideline that points to the ZigBee Health Care (HC) Profile version 1.0 [ZigBee HCP]. All subsequent requirements in Clause 10.2.1 refer to this specification. Because the commissioning of ZigBees can be challenging, in particular for large-scale networks due to the wireless nature of the connections, it is important to specify the proper procedures for the commissioning of ZigBee PHDs, which include network-joining and application pairing of devices, and device discovery, as well as security mechanisms. It is equally important to inform the users and installers of relevant events related to commissioning, such as the successful application pairing of PHDs, and the reasons for failure. These required procedures and notifications are defined in the ZigBee Health Care Profile version 1.0.

Table 10-1 – ZigBee health care profile map

Name	Description	Comments
ZigBee-HC-Map	Continua ZigBee service and client components **shall** implement ZigBee Health Care Profile version 1.0 subject to the design guidelines below	

10.2.1.2 Quality of service

The following requirements describe how QoS attributes are used for Continua ZigBee components.

Table 10-2 – ZigBee quality of service

Name	Description	Comments
ZigBee-QoS-Best.Medium	Continua ZigBee service and client components that implement the Continua *best.medium* QoS bin **shall** utilize ZigBee APS acknowledgements	
ZigBee-QoS-Good.Medium	Continua ZigBee service and client components that implement the Continua *good.medium* QoS bin **shall not** utilize ZigBee APS acknowledgements	

10.2.1.3 ZigBee multiple connections

The following requirements describe how the ZigBee health care profile is used for multiple concurrent ZigBee interface connections.

Table 10-3 – ZigBee multiple connections

Name	Description	Comments
ZigBee-MultipleConnections	Continua ZigBee service components that establish multiple ZigBee interface connections as described in Clause 10.2.2.1 **shall** use a separate ZigBee endpoint for each	

10.2.2 ZigBee data/messaging layer

This clause contains data/messaging layer design guidelines that are specific to the ZigBee interface, and thus it is not part of the set of common data/messaging layer design guidelines in Clause 6.2.

10.2.2.1 ZigBee component one-to-many connectivity

This clause describes guidelines for a sensor entering a one-to-many connectivity relationship, i.e., a ZigBee service component establishing multiple concurrent ZigBee interface connections at the same instant in time. Example scenarios include multi-function sensors providing different functionality to multiple PHGs, as well as single-function sensors providing its single functionality to multiple PHGs at the same instant in time. How to use the ISO/IEEE 1073-20601 mechanisms for association, sensor time control and PM-store usage in a one-to-many connectivity scenario are described.

10.2.2.1.1 ZigBee dominant association

The 'dominant association' concept is introduced for managing on the service component multiple simultaneous associations with one or more client components. Only through a dominant association, is a service component granting a client component control over its clock and persistently stored data. A service component can have zero or one dominant association. By this, potential conflicts of multiple client components trying to control these resources on the agent are prevented. Client components are largely unaffected by the dominant association concept. Almost all guidelines within this clause apply to service components only.

Table 10-4 – ZigBee dominant association

Name	Description	Comments
ZigBee-11073-20601-One-to-Many-Connect	Any Continua ZigBee service component that establishes more than one, simultaneous connection to one or more ZigBee client components at the same point in time **shall** create an ISO/IEEE 11073-20601 association to a ZigBee client component per connection and follow the guidelines in the remainder of this table	This guideline provides guidance for a device to establish multiple concurrent ZigBee connections

Name	Description	Comments
ZigBee-11073-20601-One-to-Many-SinglePHG	A Continua ZigBee service component that connects to a single ZigBee client component **may** create a single connection or multiple connections for providing its functions	The use of multiple connections allows turning on and off the connection of individual functions of the agent without affecting the connection of the other functions. However, in some cases, using a single connection only can be required, e.g. in case the ZigBee client component rejects the request for more than a single connection due to the fact that it is compliant to the 2010 CDG release and does not expect multiple connection requests from a single ZigBee service component
ZigBee-11073-20601-One-to-Many-ConnectionSetup	Continua ZigBee service components that establish more than one, simultaneous connection to one ZigBee client components at the same point in time **shall** create a new association to that ZigBee client component, if and only if, all other connections are in the *Unassociated* or *Operating* state	This guideline ensures that connection set-up is completed before the creation of an additional connection, and thus reduces unnecessary complexity on the client side to deal with multiple associations simultaneously
ZigBee-11073-20601-DominantAssoc	Continua ZigBee service components **shall** have at most a single dominant ISO/IEEE 11073 association at a single point in time	A ZigBee service component provides the PHG control of its resources (e.g. setting of real time clock and removal of PM-Store data) via its dominant association only. An ISO/IEEE 11073 association becomes the dominant association if one or more of the following MDS-Time-Info attribute bits or PM-Store-Capab attribute bits are set: *mds-time-mgr-set-time, mds-time-capab-set-clock, pmsc-clear-segm-by-list-sup, pmsc-clear-segm-by-time-sup, pmsc-clear-segm-remove, pmsc-clear-segm-all-sup*
ZigBee-11073-20601-DominantAssoc-ControlBits	Continua ZigBee service components **shall not** set any of following MDS-Time-Info attribute bits or PM-Store-Capab attribute bits for other than its dominant association: *mds-time-mgr-set-time, mds-time-capab-set-clock, pmsc-clear-segm-by-list-sup, pmsc-clear-segm-by-time-*	

Name	Description	Comments
	sup, pmsc-clear-segm-remove, pmsc-clear-segm-all-sup	
ZigBee-11073-20601-DominantAssoc-SetTime	Continua ZigBee service components that modified their clock based on the reception of a Set-Time action via its dominant association **shall** send an event report that contains the new *Date-and-Time* attribute value for all their non-dominant associations prior to sending any temporarily stored measurements and prior to starting a new transfer of a PM-Segment	In case the service component receives the Set-Time action during an ongoing PM-Segment transfer, see ZigBee-11073-20601-DateAndTimeUpdate-PMSegmentTransfer-* for further guidance
ZigBee-11073-20601-DominantAssoc-Closing	Continua ZigBee service components **may** close their dominant association	
ZigBee-11073-20601-DominantAssoc-Downgrading	Continua ZigBee service components **may** downgrade their dominant association to become a non-dominant association	Downgrading of the dominant association to a non-dominant association is achieved by sending an event report containing corresponding updates for the MDS-Time-Info attribute bits, so that the conditions of ZigBee-11073-20601-DominantAssoc-ControlBits for non-dominant associations are met. Note that the PM-Store-Capab attribute is static. Changing its bit values requires releasing the association and associating again, using a different configuration
ZigBee-11073-20601-DominantAssoc-Upgrading	Continua ZigBee service components that do not have a dominant association **may** upgrade an existing non-dominant association to become the dominant association	Upgrading an existing association to a dominant association is achieved by sending an event report containing corresponding updates for the MDS-Time-Info attribute bits. Note that the PM-Store-Capab attribute is static. Changing its bit values requires releasing the association and associating again, using a different configuration

10.2.2.1.2 ZigBee Time-stamping

This clause describes additional requirements for the use of time stamps as specified in [IEEE 11073-20601].

Table 10-5 – ZigBee Time-stamping

Name	Description	Comments
ZigBee-11073-20601-DataDuplicate-Timestamping	Continua ZigBee service components **shall** time-stamp data that is intended to be sent multiple times, over different connections	Sending the same data multiple times can be done over the same connection or over different connections. If time stamps were missing and if the same data was sent multiple times over different connections to separate PHGs, then those PHGs would be responsible for time-stamping and might have different notions of time. To cover scenarios like this, this guideline sets more restrictions for the time-stamping of data sent multiple times. According to [IEEE 11073-20601] data needs to be time-stamped only if it is locally stored or persistently stored on an agent before being transmitted
ZigBee-11073-20601-FixedTimeStamps	Continua ZigBee service components **shall** use the same time stamp for data that is transmitted multiple times	An example scenario where this guideline applies is the case that a service component sends the same data to multiple different clients and assigns time stamps while transmitting the data instead of while sampling the data. According to this guideline, the time stamps used for the same data are required to be identical

10.2.2.1.3 ZigBee Timeout management

This clause describes additional requirements improving interoperability in cases where timeouts as specified in [IEEE 11073-20601] are not met.

Table 10-6 – ZigBee Timeout management

Name	Description	Comments
ZigBee-11073-20601-TimeoutIndication	Continua ZigBee service components **shall** not cause a timeout on a particular connection, due to activity related to another existing connection	Here, timeouts caused by service components relate to an expected response to a GET request, a confirmed SET command, or a confirmed Action command, invoked by a ZigBee client component being in the operating state
ZigBee-11073-20601-PM-Store-TransferTimeout	Continua ZigBee service components that implement and use the PM-Store model **should** correctly initialize the PM-Segment object *Transfer-Timeout* attribute to a value accounting for the maximum number of entries stored in the segment, as well as the maximum number of supported ongoing segment transfers via other associations	The size of a segment, as well as the amount of traffic due to potential concurrent segment transfer via other connections affects the time needed for transferring a complete PN-Segment

10.3 ZigBee Certified Capability Classes

Table 10-7 shows the certified Capability Classes defined for the ZigBee interface design guidelines. A certification program run by Continua Health Alliance exists for devices that implement the CDG. For ZigBee PHDs and PHGs, the certification testing will be performed on an integrated device, meaning the testing and certification is applied to the hardware and software of the device. Changes to components of the device may require a re-certification.

Table 10-7 also references the guidelines (clause numbers) that are applicable for each of the certified Capability Classes on the service as well as the client side.

Table 10-7 – ZigBee Certified Capability Classes

Certified Capability Classes	Relevant guidelines
ZigBee activity hub service, ZigBee activity hub client	6.2, 6.3.14, 10.2
ZigBee adherence monitor service, ZigBee adherence monitor client	6.2, 6.3.29, 10.2
ZigBee basic 1-3 lead ECG service, ZigBee basic 1-3 lead ECG client	6.2, 6.3.2, 10.2
ZigBee blood pressure monitor service, ZigBee blood pressure monitor client	6.2, 6.3.4, 10.2
ZigBee body composition analyser service, ZigBee body composition analyser client	6.2, 6.3.9, 10.2
ZigBee cardiovascular fitness service, ZigBee cardiovascular fitness client	6.2, 6.3.11, 10.2
ZigBee cardiovascular step counter service, ZigBee cardiovascular step counter client	6.2, 6.3.12, 10.2

Certified Capability Classes	Relevant guidelines
ZigBee CO sensor service, ZigBee CO sensor client	6.2, 6.3.27, 10.2
ZigBee contact closure sensor service, ZigBee contact closure sensor client	6.2, 6.3.18, 10.2
ZigBee continuous glucose monitor service, ZigBee continuous glucose monitor client	6.2, 6.3.31, 10.2
ZigBee dosage sensor service, ZigBee dosage sensor client	6.2, 6.3.20, 10.2
ZigBee enuresis sensor service, ZigBee enuresis sensor client	6.2, 6.3.17, 10.2
ZigBee fall sensor service, ZigBee fall sensor client	6.2, 6.3.15, 10.2
ZigBee gas sensor service, ZigBee gas sensor client	6.2, 6.3.28, 10.2
ZigBee glucose meter service, ZigBee glucose meter client	6.2, 6.3.7, 10.2
ZigBee heart-rate sensor service, ZigBee heart-rate sensor client	6.2, 6.3.3, 10.2
ZigBee INR meter service, ZigBee INR meter client	6.2, 6.3.8, 10.2
ZigBee insulin pump service ZigBee insulin pump client	6.2, 6.3.32, 10.2
ZigBee motion sensor service, ZigBee motion sensor client	6.2, 6.3.16, 10.2
ZigBee pulse oximeter service, ZigBee pulse oximeter client	6.2, 6.3.1, 10.2
ZigBee peak flow monitor service, ZigBee peak flow monitor client	6.2, 6.3.10, 10.2
ZigBee PERS sensor service, ZigBee PERS sensor client	6.2, 6.3.26, 10.2
ZigBee property exit sensor service, ZigBee property exit sensor client	6.2, 6.3.23, 10.2
ZigBee smoke sensor service, ZigBee smoke sensor client	6.2, 6.3.22, 10.2
ZigBee strength fitness service, ZigBee strength fitness client	6.2, 6.3.13, 10.2
ZigBee switch sensor service, ZigBee switch sensor client	6.2, 6.3.19, 10.2
ZigBee temperature sensor service, ZigBee temperature sensor client	6.2, 6.3.24, 10.2
ZigBee thermometer service, ZigBee thermometer client	6.2, 6.3.5, 10.2
ZigBee usage sensor service, ZigBee usage sensor client	6.2, 6.3.25, 10.2

Certified Capability Classes	Relevant guidelines
ZigBee water sensor service, ZigBee water sensor client	6.2, 6.3.21, 10.2
ZigBee weighing-scales service, ZigBee weighing-scales client	6.2, 6.3.6, 10.2

11 Bluetooth Low Energy design guidelines

11.1 Architecture (informative)

The Bluetooth Low Energy protocol, also known as Bluetooth Smart, is also a Continua supported transport technology for the |PHD-IF as a widely supported low-energy, low-bandwidth, limited range wireless protocol. The Bluetooth Special Interest Group (SIG) has defined device specific profiles and services on top of the Bluetooth Low Energy Attribute Profile that are supported by the PHD-IF. This is illustrated in Figure 11-1.

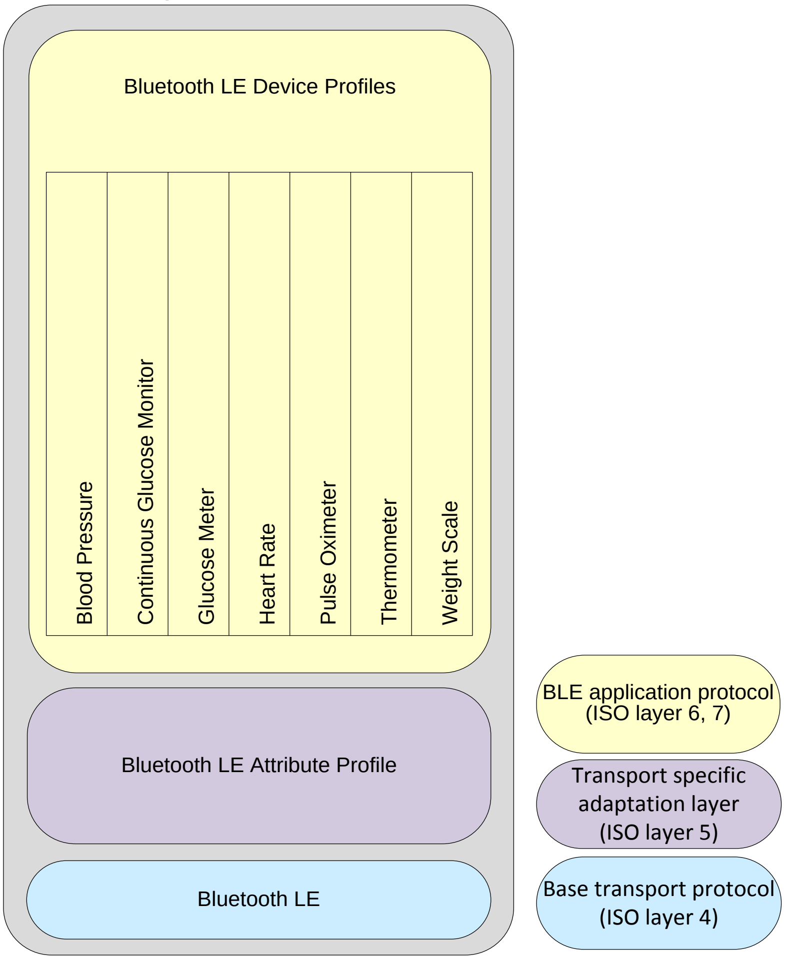

Figure 11-1 – Bluetooth LE interface stack

The Bluetooth LE interface does not utilize the [IEEE 11073-20601] protocol for data exchange. The Bluetooth LE interface utilizes the Bluetooth LE protocol with data types compatible to the IEEE 11073-10101 nomenclature and the IEEE 11073-20601 domain information model. For the characteristics defined in the Bluetooth low energy profiles, the *Personal Health Devices Transcoding White Paper* describes how to transcode into an equivalent IEEE DIM and/or nomenclature representation. At a minimum, this covers the mandatory attributes from the supported [ISO/IEEE 11073-104xx] device specializations.

The following Bluetooth LE device-specific specifications from the Bluetooth SIG apply to the Bluetooth LE interface.

– Health thermometer profile and service (e.g. temperature)
– Heart rate profile and service (e.g. heart rate, R-R interval)
– Device information service (e.g. manufacturer name, model number, serial number, hardware revision, firmware revision, software revision, system ID)

- Blood pressure profile and service (e.g. blood pressure measurement, intermediate cuff pressure)
- Glucose profile and service (e.g. glucose measurement)
- Weight Scale profile, Weight Scale service and Body Composition services (e.g. weight measurement, BMI, body fat mass percentage).
- Continuous Glucose Monitor profile and service (e.g. glucose measurements)
- Pulse Oximeter profile and service (e.g. SpO2 measurement)
- Personal Health Devices Transcoding White Paper describes how to transcode Bluetooth low energy data structures and format into an equivalent IEEE 11073 PHD data representation regarding DIM and/or nomenclature

11.2 Bluetooth LE interface guidelines

11.2.1 Bluetooth LE services and profiles

Bluetooth LE technology has been selected as the low-power (LP) wireless technology. The specifications relating to Bluetooth LE are in version 4.0 (or later) of the core Bluetooth specifications [Bluetooth CS4.0]. Any related profile specifications are detailed in separate documents. Bluetooth PHDs and PHGs that support Bluetooth LE can be either a dual mode device, which is a device that supports both standard BR/EDR Bluetooth and Bluetooth LE, or a single mode device, which is a device that supports Bluetooth LE only. It is envisioned that service components supporting Bluetooth LE will mostly be single mode devices.

Table 11-1 – Bluetooth LE transport

Name	Description	Comments
Bluetooth-LE-Map	Continua Bluetooth LE service and client components **shall** implement Bluetooth LE as described in Bluetooth *Core Version 4.0* [Bluetooth CS4.0] subject to the design guidelines below.	Note that later backwards-compatible versions of the Bluetooth low energy specification can also be used to meet this requirement.

11.2.2 Device discovery, pairing and service discovery

Continua Bluetooth LE service devices (PHDs) transfer measurement data to client devices (PHGs). Continua Bluetooth LE client and service components are required to pair with each other, either following a search initiated by the client component that obtains a list of compatible devices or through an out-of-band configuration.

A process of discovery of the service component by the client component is required for all Continua Bluetooth LE devices. This ensures a consistent and user-friendly pairing procedure.

The guidelines throughout this clause create a single and universally supported technique for pairing devices that give a minimum of surprise or inconvenience to users.

Table 11-2 – Bluetooth LE device discovery, pairing and service discovery

Name	Description	Comments
Bluetooth-LE-Pairing-Start-Client	Once a Continua Bluetooth LE client component has discovered a Continua Bluetooth LE service component that supports a compatible service, it **shall** support pairing with that Continua Bluetooth LE service component.	
Bluetooth-LE-Enter-Discoverability-Service	A Continua Bluetooth LE service component **shall** have a documented way to be set to be discoverable and a documented way to pair with a Continua Bluetooth LE client component.	
Bluetooth-LE-Initiate-Discovery-Pairing-Client	A Continua Bluetooth LE client component **shall** have a documented way to initiate a search for discoverable Continua Bluetooth LE service component and a documented way of initiating pairing with a Continua Bluetooth LE service component.	
Bluetooth-LE-Discoverability-Mode-Service	A Continua Bluetooth LE service component **shall not** be discoverable unless initiated by a user.	
Bluetooth-LE-Delete-Pairing-Service	A Continua Bluetooth LE service component **should** have a way to delete pairings.	
Bluetooth-LE-Delete-Pairing-Client	A Continua Bluetooth LE client component **should** have a way to delete pairings.	
Bluetooth-LE-Additional-Pairing-Service	A Continua Bluetooth LE service component **shall** support replacing its pairing.	Pairing is not exclusive for the lifetime of the service component to enhance interoperability
Bluetooth-LE-No-Data-Exchange-Before-Pairing-Service	Continua Bluetooth LE service component data (other than service discovery data or capability or service name from the advertising packet) **shall not** be exchanged with a Continua Bluetooth LE client component prior to pairing.	

Name	Description	Comments
Bluetooth-LE-Disc-Mode-Max-Duration-Service	A Continua Bluetooth LE service component **should** have a documented maximum duration for discoverable mode whereby after the maximum time, the Continua Bluetooth LE service component ceases to be discoverable until put back into that mode by the user.	
Bluetooth-LE-After-Pairing-Undiscoverable-Service	After a Continua Bluetooth LE service component is successfully paired, it **shall** immediately (e.g. within 1 second) become undiscoverable until made discoverable again by the user.	
Bluetooth-LE-Store-Pairing-Service	Continua Bluetooth LE service components **should** store pairing data from at least the most recently paired device such that the data is persistent (e.g. with loss of power, including removal of a battery).	
Bluetooth-LE-Store-Pairing-Client	Continua Bluetooth LE client components **should** store pairing data from at least the most recently paired device such that the data is persistent (e.g. with loss of power including removal of a battery).	
Bluetooth-LE-Number-Store-Pairing-Client	Continua Bluetooth LE client components **should** store pairing data for at least the number of devices they are intended to simultaneously support.	
Bluetooth-LE-Supported-Services-Profiles-Service	Continua Bluetooth LE service component's Attribute database **shall** list all supported Bluetooth LE services/profiles claimed in Continua certification documentation.	

11.2.3 User notification

Establishing a new pairing relationship is an important event. Because of the potential for confusion, extreme care should be used before automating the pairing procedure. To allow users reasonable control of their CDG systems, PHGs are required to provide a facility for alerting users of significant events. Because discovery may be difficult for users to understand, it is important to inform them of new pairings and reasons for failure. The guidelines in this clause intentionally leave the nature of notifying and informing the user to be defined by the manufacturer.

Table 11-3 – Bluetooth LE user notification

Name	Description	Comments
Bluetooth-LE-Inform-Pairing-Success-Service	If supported by the UI, Continua Bluetooth LE service components **should** inform the user that pairing and authentication was successful.	
Bluetooth-LE-Inform-Pairing-Success-Client	If supported by the UI, Continua Bluetooth LE client components **shall** inform the user that a pairing and authentication was successful.	
Bluetooth-LE-Filter-Compatible-Client	Continua Bluetooth LE client components in a mode of device discovery **should** filter discovered Continua Bluetooth LE service components to include only those that have compatible services/profiles.	
Bluetooth-LE-Inform-User-Pairing-Failure-Client	If there is a failure during the discovery, pairing and authentication process, and if supported by the UI, the Continua Bluetooth LE client component **shall** inform the user whether the failure is because 1) no compatible Continua Bluetooth LE service components was found (compatible device not found) or 2) the pairing failed (pairing failure) or 3) the authentication process timed out (authentication timeout) or 4) the user entered the incorrect passkey (incorrect PIN).	

11.2.4 Quality of Service

For Bluetooth LE, the QoS is defined within the applicable Bluetooth LE profile.

11.2.5 Authentication

In Bluetooth LE profiles referenced in these guidelines, the service component chooses the mode of security it desires and the client component is required to accept this. Bluetooth LE profiles can mandate Just Works authentication, Passkey Entry of a six-digit PIN or an out-of-band obtained passkey. While in Bluetooth there are various authentication options, CDG places more requirements on authentication as follows to ensure interoperability.

Table 11-4 – Bluetooth LE authentication

Name	Description	Comments
Bluetooth-LE-Authentication-Support-Service	Continua Bluetooth LE service components **shall** support at least one of the following Bluetooth 4.0 pairing methods depending on its I/O capabilities and the appropriate security for the service component device type: Just Works or Passkey Entry.	I/O capabilities include display, keyboard, yes/no. See Bluetooth Core Specification 4.0 [Bluetooth CS4.0] for further information.
Bluetooth-LE-Authentication-Support-Client	Continua Bluetooth LE client components **shall** support Just Works and Passkey Entry pairing methods for Bluetooth 4.0 if the client component has the appropriate I/O capabilities.	I/O capabilities include display, keyboard, yes/no. See Bluetooth Core Specification 4.0 [Bluetooth CS4.0] for further information. This pairing guideline is necessary to ensure interoperability and give reasonable assurance that a service component's chosen pairing method will be supported by client components.

11.2.6 OEM requirements

Bluetooth LE profiles referenced in these guidelines may define some OEM characteristics within the Bluetooth SIG device information service as optional. This clause describes the guidelines that are targeted at the OEM characteristics. All of the fields defined in this clause are from the Bluetooth SIG device information service.

Table 11-5 – Bluetooth LE OEM requirements

Name	Description	Comments
Bluetooth-LE-11073-20601-Manufacturer	Continua Bluetooth LE service components **shall** support and set the manufacturer name string defined in the Bluetooth SIG device information service to the device's original manufacturer's name. If this capability is available, the manufacturer name string **may** be overwritten to the customer facing company's name by the customer facing company.	

Name	Description	Comments
Bluetooth-LE-11073-20601-Model	Continua Bluetooth LE service components **shall** set the model number string defined in the Bluetooth SIG device information service to the device's original manufacturer's model number. The model number string field **may** be overwritten to the customer facing company's model by the customer facing company.	
Bluetooth-LE-11073-20601-SYSID	Continua Bluetooth LE service components **shall** include the System ID characteristic defined in the Bluetooth SIG device information service.	
Bluetooth-LE-11073-20601-OUI	The organizationally unique identifier (OUI) field of the System ID characteristic defined in the Bluetooth SIG device information service in a Continua Bluetooth LE service component **shall** be set and remain unchanged from the value set by the original manufacturer.	This is a unique identifier, which is obtained by the IEEE registration authority and which is associated with a company. This attribute maps to the OUI part (first 24 bits) of the EUI-64 attribute
Bluetooth-LE-11073-20601-DID	The 40 bit manufacturer defined identifier field in the System ID characteristic defined in the Bluetooth SIG device information service of a Continua Bluetooth LE service component **shall** be set and remain unchanged from the value set by the original manufacturer.	In combination with the OUI part above, this is a unique identifier associated with the device. It is required in order to facilitate data quality analysis. This attribute maps to the company defined part (last 40 bits) of the EUI-64 attribute
Bluetooth-LE-11073-20601-Serial-Number	Continua Bluetooth LE service components **shall** set the serial number string characteristic defined in the Bluetooth SIG device information service to the serial number of the device.	
Bluetooth-LE-11073-20601-FW-Revision	Continua Bluetooth LE service components that provide a firmware identifier **shall** set the firmware revision string characteristic defined in the Bluetooth SIG device information service to the firmware identifier of the device.	The firmware identifier is the version of the firmware deployed on the device. The firmware release deployed on the device is uniquely identified by the firmware identifier

11.2.7 Date and time requirements

Bluetooth LE devices which report time-stamped measurements must provide the means to report the current date and time of the device. The following guidelines are intended to provide the means for this support.

Table 11-6 –Bluetooth LE date and time requirements

Name	Description	Comments
Bluetooth-LE-Date-Time	Continua Bluetooth LE service components that report time-stamped measurements shall support the Current Time Service [Bluetooth CTS] or shall include the "Date Time" characteristic in the service component for the purpose of reporting the current date and time of the service component.	Transcoding of time specified in the personal health devices transcoding white paper from the Bluetooth SIG [Bluetooth PHDT v1.5]. Newer versions of this whitepaper require support of CTS by the service component when reporting time-stamped measurements. For newer designs the use of CTS is the preferred choice. Continua still allows use of the Date-Time characteristic for legacy devices that report time-stamped measurements as described in [Bluetooth PHDT V1.4].

11.2.8 Certification and regulatory aspects

Since Bluetooth LE profiles referenced in these guidelines define as optional the IEEE 11073-20601 Regulatory Certification Data List characteristic within the Bluetooth SIG device information service, this clause describes the guidelines that are targeted at certification and regulatory including those specific to this characteristic.
For this purpose, the ASN-1 definitions from Figure 6-3 are referenced in Table 11-7.

```
ContinuaStructType ::= INT-U8 {
     continua-version-struct(1),   -- auth-body-data is a ContinuaBodyStruct
     continua-reg-struct(2)          -- auth-body-data is a ContinuaRegStruct

}
ContinuaBodyStruct ::= SEQUENCE {
        major-IG-version      INT-U8,
        minor-IG-version      INT-U8,
        certified-devices     CertifiedCapabilityClassList
}

CertifiedCapabilityClassList ::= SEQUENCE OF CertifiedCapabilityClassEntry

-- See guideline  11073-20601-CapabilityEntry for the algorithm to compute the
value
CertifiedCapabilityEntry ::= INT-U16

ContinuaRegStruct ::= SEQUENCE {
        regulation-bit-field     RegulationBitFieldType
}
```

```
RegulationBitFieldType  ::= BITS-16 {
    unregulated-device (0)  -- This bit shall be set if the device is not
regulated }
```

Figure 11-2 – ASN.1 notation of Continua certification structures for Bluetooth LE

Table 11-7 – Bluetooth LE certification and regulation

Name	Description	Comments
Bluetooth-LE-Support-Reg-Cert-Data-Service	Continua Bluetooth LE service components **shall** support and fill the IEEE 11073-20601 Regulatory Certification Data List characteristic defined in the Bluetooth SIG device information service with an MDER encoded version of the IEEE 11073-20601 RegCertDataList data structure. The RegCertDataList data structure **shall** contain a RegCertData element with the *auth-body-continua* and the *auth-body-struc-type* field set to *continua-version-struct* from a ContinuaStructType as defined above. The field *auth-body-data* **shall** be filled in as a *ContinuaBodyStruct* as defined above in Figure 11-2.	This is used to indicate whether a device is Continua certified and (if so) which version of the Continua Design Guidelines it is certified to
Bluetooth-LE-CapabilityList	Continua Bluetooth LE service components **shall** list all implemented and only the implemented certified Capability Classes in the IEEE 11073-20601 Regulatory Certification Data List characteristic within the Bluetooth SIG device information service.	

Name	Description	Comments
Bluetooth-LE-CapabilityEntry	Continua Bluetooth LE service components **shall** assign the following certified Capability Class field value within the IEEE 11073-20601 Regulatory Certification Data List characteristic within the Bluetooth SIG device information service to an implemented Certified Capability Class: MDC_DEV_*_SPEC_PROFILE_* - 4096 + TCode x 8192, where MDC_DEV_*_SPEC_PROFILE_* denotes the IEEE 11073 PHD nomenclature code for the corresponding device (sub-) specialization, and TCode denotes the corresponding transport standard, with TCode = {4 for LP wireless PAN}	See [Bluetooth PHDT]
Bluetooth-LE-Report-Regulated-Service	All Continua BLe service components **shall** report information on whether or not they are regulated. This is a single Boolean entitled unregulated-device, which is set to 1 if not regulated and 0 if regulated and contained as part of IEEE 11073-20601 Regulatory Certification Data List defined in the Bluetooth SIG device information service.	

11.2.9 Transcoding

Bluetooth LE profiles referenced in these guidelines are designed to be compatible with the IEEE 11073 device information model (DIM) and nomenclature of a corresponding IEEE 11073 device specialization. The Bluetooth SIG published document [Bluetooth PHDT] contains the information showing how the applicable Bluetooth LE characteristics can be mapped to the device information model (DIM) and nomenclature of the corresponding IEEE 11073 device specializations. From a Bluetooth LE profile perspective this mapping information is included as informative text for profiles targeted for usage in CDG. However, when Bluetooth LE profiles are used within the CDG and transcoding is required, this mapping information is normative for implementations that transcode Bluetooth LE data.

Table 11-8 – Bluetooth LE transcoding

Name	Description	Comments
Bluetooth-LE-Transcode	The guidelines for interfaces in the Continua E2E architecture assume that data coming from the X73 interface are IEEE 11073 nomenclature and DIM representations and then specify necessary data conversions for each of the interfaces. Any solution that interacts with the Bluetooth LE interface and passes the data over other Continua interfaces **shall** follow [Bluetooth PHDT] during the translation process from Bluetooth LE data to final representation for the supported interface(s). Transcoded data **shall** be compliant to the IEEE 11073 nomenclature and DIM corresponding specifically with [IEEE 11073-20601].	[Bluetooth PHDT] is informative from the Bluetooth SIG perspective, but is normative for the purposes of these guidelines. This white paper specifies how to convert the Bluetooth LE data into full IEEE 11073 compliant data, which then supports the use of the data for the Continua Services and HIS Interfaces. Note that this guideline does not require a PHG to actually create the DIM, objects, and attributes indicated by the white paper. However, the data generated for transmission over the subsequent Continua interface must match the data that would have been generated from such a DIM

11.3 Bluetooth LE PHDs and PHGs

11.3.1 Blood pressure monitor

Table 11-9 – Blood pressure general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-Blood Pressure-Service	Continua Bluetooth LE blood pressure service components **shall** implement the blood pressure sensor role as defined by the Blood Pressure profile and service – [Bluetooth BPP] and [Bluetooth BPS].	
Bluetooth-LE-Blood Pressure-Client	Continua Bluetooth LE blood pressure client components **shall** implement the collector role as defined by Blood Pressure profile from [Bluetooth BPP].	

11.3.2 Thermometer

Table 11-10 – Thermometer general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-Thermometer-Service	Continua Bluetooth LE thermometer service components	

Name	Description	Comments
	shall implement the health thermometer sensor role as defined by the Health Thermometer prodfile and service – [Bluetooth HTP] and [Bluetooth HTS].	
Bluetooth-LE-Thermometer-Client	Continua BLe thermometer client components **shall** implement the collector role as defined by the Health Thermometer profile from [Bluetooth HTP].	

11.3.3 Heart-rate sensor

Table 11-11 – Heart-rate sensor general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-Heart-rate-Sensor-Service	Continua Bluetooth LE heart-rate sensor service components **shall** implement the heart-rate sensor role as define by the Heart Rate profile and service –[Bluetooth HRP] and [Bluetooth HRS].	
Bluetooth-LE-Heart-Rate-Sensor-Client	Continua Bluetooth LE heart-rate client components **shall** implement the collector role as defined by the Heart Rate profile from [Bluetooth HRP].	

11.3.4 Glucose meter

Table 11-12 – Glucose meter general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-Glucose-Meter-Service	Continua Bluetooth LE glucose meter service components **shall** implement the glucose sensor role as defined by the Glucose profile and service – [Bluetooth GLP] and [Bluetooth GLS].	
Bluetooth-LE-Glucose-Meter-Client	Continua Bluetooth LE glucose meter client components **shall** implement the collector role as defined by the Glucose Meter profile from [Bluetooth GLP].	

11.3.5 Weight Scale

Table 11-13 – Weight Scale general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-Weight-Scale-Service	Continua Bluetooth LE weight scale meter service components **shall** implement the weight scale sensor role as defined by the Weight Scale profile and service from the Bluetooth SIG - [Bluetooth WSP] and [Bluetooth WSS].	
Bluetooth-LE-Weight-Scale-Body-Composition-Service	Continua Bluetooth LE weight scale service components **may** implement the body composition service from the Bluetooth SIG [Bluetooth BCS].	
Bluetooth-LE-Weight-Scale-Client	Continua Bluetooth LE weight scale client components **shall** implement the collector role as defined by the the Weight Scale profile from the Bluetooth SIG [Bluetooth WSP].	

11.3.6 Continuous Glucose Monitor

Table 11-14 – CGM general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-CGM-Service	Continua Bluetooth LE CGM service components **shall** implement the CGM sensor role as defined by the CGM profile and service - [Bluetooth CGMP] and [Bluetooth CGMS].	
Bluetooth-LE-CGM-Client	Continua Bluetooth LE CGM client components **shall** implement the collector role from the CGM profile from the Bluetooth SIG [Bluetooth CGMP].	

11.3.7 Pulse Oximeter

Table 11-15 – Pulse Oximeter general requirements for Bluetooth LE

Name	Description	Comments
Bluetooth-LE-POX-Service	Continua Bluetooth LE Pulse Oximeter service components **shall** implement the Pulse Oximeter sensor role as defined	

Name	Description	Comments
	by the Pulse Oximeter profile and service - [Bluetooth POXP] and [Bluetooth POXS].	
Bluetooth-LE-POX-Client	Continua Bluetooth LE Pulse Oximeter client components **shall** implement the collector role from the Pulse Oximeter profile from the Bluetooth SIG [Bluetooth POXP].	

11.4 Bluetooth LE Certified Capability Classes

Table 11-16 shows the certified Capability Classes defined for the Bluetooth LE interface design guidelines. A certification program run by Continua Health Alliance exists for PHDs and PHGs that implement the CDG. For Bluetooth LE PHDs and PHGs, the certification testing will be performed on an integrated device, meaning the testing and certification is applied to the hardware and software of the device. Changes to components of the device may require a re-certification. Table 11-16 also references the guidelines that are applicable for each of the certified Capability Classes.

Table 11-16 – Bluetooth LE Certified Capability Classes

Certified Capability Classes	relevant guidelines
Bluetooth LE blood pressure monitor service Bluetooth LE blood pressure monitor client	11.2, 11.3.1
Bluetooth LE continuous glucose monitor service Bluetooth LE continuous glucose monitor client	11.2, 11.3.6
Bluetooth LE glucose meter service Bluetooth LE glucose meter client	11.2, 11.3.4
Bluetooth LE heart-rate sensor service Bluetooth LE heart-rate sensor client	11.2, 11.3.3
Bluetooth LE pulse oximeter service Bluetooth LE pulse oximeter client	11.2, 11.3.7
Bluetooth LE thermometer service Bluetooth LE thermometer client	11.2, 11.3.2
Bluetooth LE weighing-scales service Bluetooth LE weighing-scales client	11.2, 11.3.5

Appendix I Additional Bluetooth BR/EDR Information

(This appendix does not form an integral part of these guidelines)

I.1 Bluetooth terminology

BR/EDR: Abbreviation for Basic Rate/Enhanced Data Rate. BR/EDR is usually used as a way to describe "Classic" Bluetooth, as opposed to Bluetooth high speed or Bluetooth low energy.
Discoverable: A Bluetooth device is discoverable if it is periodically entering the Inquiry Scan substate. Inquiry Scan requires an active receiver for about 11.25 ms (default), and is entered at least once every 2.56 s. If a device is discoverable, it will respond to Inquiry procedures (usually a general Inquiry) from any device that wants to search.
Connectable: A Bluetooth device is connectable if it is periodically entering the Page Scan substate. Page Scan requires an active receiver for about 11.25 ms (default), and can be entered continuously or periodically. Normal periods are in the one second range (modes R2 <= 2.56 s, R1 <= 1.28 s, R0 is continuous). If a device is connectable, it will respond to pages from devices that address it specifically (by Bluetooth MAC).
Limited Discoverable: A Bluetooth term for devices that are sometimes discoverable, and sometimes not.
Discovery: Using the Inquiry substate to learn of the existence of other Bluetooth devices within transmission range. May take up to thirty seconds. Sometimes called "device discovery" to distinguish from service discovery.
Pairing: Exchanging link keys to establish a future trust relationship with a known device. Performed with Secure Simple Pairing (SSP), except in legacy cases.
Service Discovery: Creating a baseband connection to a specific device (may be paired, but does not need to be) to discover details about services offered on that device.
Out-of-Band Connection: A data link other than the Bluetooth connection. This may include Bluetooth near-field communication (NFC), patch cables, removable media, or any other mechanism for transferring data between the two devices.

I.2 Bluetooth BR/EDR pairing methods

Starting with Bluetooth 2.1+EDR, pairing uses "Secure Simple Pairing" (SSP) which (as the name implies) improved both the security and the simplicity of the Bluetooth pairing procedure. Older devices use a legacy pairing procedure. Both of these procedures result in a shared "link key" that is unique to the pair of devices, and can be used both to authenticate future connections and to create session keys for encrypting traffic over the air.
Whichever procedure is used, the user experience will depend heavily on how it is implemented. To produce an adequate level of trust between the two devices while also giving a good user experience, the following factors are particularly relevant:
Security against eavesdropping refers to the required protection from listening devices that are present during the pairing procedure. Legacy pairing offers moderate protection only if long PINs are used (at least six digits), although attacks are still possible. SSP is always secure against eavesdropping.
Security against Active Man-In-The-Middle (MITM) refers to the required protection from a device that inserts itself between the two parties on the physical link, so instead of pairing with each other (as intended), they both pair with the attacker. The attacker may relay data as if the connection were working correctly, but would be able to intercept or even change that data during transmission. Legacy pairing is not secure against this type of attack. SSP may be secure against it.
Security against confusion refers to the required protection against allowing a device to pair with a device other than the intended partner.

For additional information on Bluetooth discovery and pairing, including device user interface input/output capabilities, see the following Bluetooth SIG documentation as formally referenced in Clause 2 and the Bibliography of [H.810].

- Bluetooth Core Specification, v2.1 or later, Vol. 3, Part C: Generic Access Profile [Bluetooth CS2.1]
- Bluetooth Discovery White Paper [b-Bluetooth Discovery]
- Bluetooth Secure Simple Pairing User Terminology White Paper [b-Bluetooth SSP UT]
- Bluetooth User Interface Flow Diagrams for Bluetooth Secure Simple Pairing Devices White Paper [b-Bluetooth SSP UI]
- Bluetooth Secure Simple Pairing Usability Metric White Paper [b-Bluetooth SSP UM]

I.3 Bluetooth BR/EDR legacy pairing procedures

Legacy pairing requires keys from both devices. If a device has a user interface, a unique PIN can be entered. It is not recommended that well-known values (like "0000") are used for groups of devices, as this may cause erroneous pairings. PINs should be at least six digits long, and selected in such a way that each individual PIN will be re-used only about once in 1000000 devices (or less). The PIN for each device should be clearly identified on the device packaging, although that identification may be made removable.

I.4 Supporting Bluetooth OEM subsystems and components

The Bluetooth SIG currently allows the certification of "profile subsystems" devices that completely implement a profile, but are not themselves an "End Product". It is expected that some implementers will develop and market HDP modules that include the entire HDP implementation with the exception of the ISO/IEEE 11073-20601 data layer and ISO/IEEE 11073-104xx device specializations. Others may develop the ISO/IEEE 11073-20601 data layer and device specializations such that when the two implementations are combined, they form an End Product. The Bluetooth Qualification System allows for two partial implementations to be combined forming an "End Product" through the combination of appropriate subsystems or through the use of "subsetting". However, some testing of the combined implementations may be required. Refer to the Bluetooth SIG for further information regarding the Bluetooth qualification process.

I.5 Quality of service bins for Bluetooth

For Bluetooth, the expected Quality of Service (QoS) for a data connection is identified through the use of the two recognized QoS bins (see Clause 9.2.7). Achieving this QoS (knowing what is expected from a channel, policing what is being delivered, and flagging exceptional situations) is the responsibility of both ends of the connection.

In the case where a connection is point-to-point, this can often be delegated to the underlying transport layer implementation. For example, when a Bluetooth connection is established between two devices (by a successful pairing procedure), the Link Manager Protocol can request the "supported features" of the partner device. These features would include information about which enhanced data rate modes are supported, and therefore allow the local device (which already knows its own capabilities) to make a good guess at the throughput it can expect over that link. This is the recommended method for this version of the Continua Design Guidelines.

When the data is routed via intermediate nodes, but the QoS is important from end-to-end, some higher-layer function is required to accumulate and correlate the QoS expected from the various components, or at least to assign expected bounds to each hop. This will require communication of QoS characteristics at the end-to-end (transport layer). This version of the CDG support, at maximum, two cascaded transport technologies: USB/Bluetooth and ZigBee. The overall end-to-end latency is statically managed by splitting the end-to-end transport latency budget between these two transports.

See Clause 6.1.6.2 in [H.810] for a definition of the QoS bins supported by these Guidelines.

The two channel types provided for in the Bluetooth HDP specification [Bluetooth HDPv1.1] are reliable and streaming. On the reliable channel, latency will be most sensitive to retransmission times. On the streaming channel (which never retransmits data), it will be most sensitive to buffer sizes and local latency. A 10% margin is reasonable to include when making latency calculations to account for the software latency for handling of messages. The latency expected on the streaming channel can be calculated from the poll interval taking software latency into consideration.

The poll interval is the maximum number of slots that will normally be allowed to separate consecutive opportunities for a slave to begin a transmission. A slave may request a new poll interval from the master (by sending an LMP_quality_of_service_req packet) and will be informed of its value. However, the master sets that value. Legal values are any even number of slots in the range 6 through 4096 (3.75 ms - 2.56 s) and the default value is 40 (25 ms).

The streaming channel may be configured to have a polling interval short enough that, when combined with the actual transmission duration, will provide "Low" latency. However, in some particular configurations this may not be possible. For example, if the device is itself a slave, and connects to a master that does not support polling intervals other than the default, it may have the opportunity to start a new data packet only once every 25 ms.

"Medium" or longer latency should always be possible (for reasonable packet sizes) on the streaming channel.

Latency on the reliable data channel depends on retransmission. If an out-of-sequence packet is received, it will trigger retransmission of the intervening lost packets reasonably quickly. In the worst case, however, the last packet of the message may be lost (for example, if only one L2CAP packet were transmitted). In this case, retransmission would not occur until the retransmission timeout period had elapsed. This time is communicated in the option configuration information for L2CAP Enhanced Retransmission mode option and may be in the hundreds of milliseconds range. If the retransmit timer expires in the sending device and unacknowledged frames exist, they will be retransmitted.

Over a normal connection, loss of the same packet twice should be unusual, so a reliable connection should be able to deliver an average latency in the "Medium" range, if its retransmission timeout is around 100ms. Setting the MaxTransmit value to 2 would require the connection to be closed if the same packet were ever lost twice. However, very few scenarios would benefit from using this feature and MaxTransmit should usually be larger than 2.

For reliability, the Bluetooth channel has a basic bit error rate of less than 0.1%, and the data packets are protected with a 16-bit CRC. The SDU (recombined higher-layer data packet) is further protected by another 16-bit CRC (the FCS). This is true on both the reliable and streaming channels, so the probability of a bit-error in any packet should be less than 10^{-9}.

The streaming channel may lose packets (particularly due to buffer overflows) but the reliable channel will not lose packets.

Either channel may be broken due to range or extreme interference. Neither the Bluetooth Health Device Profile, nor these guidelines currently require devices to seek a reconnection following an unintentional disconnect, although the possibility is provided for in the protocols.

Before committing to an upper layer that any of these QoS bins is supported by a particular channel, an implementation shall check the relevant configuration parameters of the actual L2CAP channel (once it is established) to verify its commitment is supported.

Appendix II Additional ZigBee information

(This appendix does not form an integral part of these guidelines)

II.1 ZigBee networking

The 802.15.4/ZigBee network provides facilities for commissioning, data transfer and maintenance. Use of a certified ZigBee platform provides a robust self-healing mesh network. The ZigBee Health Care Profile mandates use of the 11073 Protocol Tunnel, and reuses components of the ZigBee Cluster Library.

Commissioning details depend on the deployment scenario. Three deployment scenarios are addressed by this profile, as follows.

1. Service provider scenario. In this scenario, a service provider that provides patient monitoring services is responsible for providing all the devices that are part of the network, and preloading these devices with all the information that they need to securely join the network and work together.
2. In-house commissioning scenario. In this scenario, the network owner (e.g. a medical care facility) has its own in-house commissioning facility, to configure the devices with all the information that they need to securely join the network and work together.
3. Consumer scenario. This scenario covers the case of small networks, where the network owner does not have a service provider, and wishes to purchase devices from multiple providers and install them himself. This case is typical of the home environment.

For example, in the consumer scenario, a typical deployment may be as follows:

1. The Coordinator or router sends a command to the ZigBee network to allow joining of new device for a limited period.
2. A ZigBee healthcare device will first do a scan for networks and build a list of available networks that allow joining.
3. The ZigBee healthcare device will then pick a network and associate to the nearest node(router or coordinator) that allows joining and start the security authentication process.
4. The router/coordinator parent will now send an update-device (device joined) message to the ZigBee security Trust Center in encrypted form.
5. The Trust Center will now determine if it will allow the device in the network or not.
6. If the device is allowed in the network the Trust Center will send the network security key to the device. Note this is done using a predefined link key.
7. The device is now an active participant in the network.

II.2 ZigBee pairing process/service discovery types

A ZigBee device consists of one or more ZigBee device descriptions (e.g. thermometer and pulse oximeter) and their corresponding application profile(s), optionally on a separate endpoint, that share a single physical IEEE802.15.4 radio. Each device has a unique 64-bit IEEE address and contains a collection of clusters and associated functionality implemented on a ZigBee endpoint. Device descriptions are defined in the scope of the ZigBee Health Care application profile. Each device description has a unique identifier that is exchanged as part of the discovery process.

The ZigBee specification [ZigBee Spec] provides the facility for devices to find out information about other nodes in a network, such as their addresses, which types of applications are running on them, their power source and sleep behaviour. This information is stored in descriptors on each node, and is used by the requesting node to tailor its behaviour to the requirements of the network. Discovery is typically used when a node is being introduced into a Health Care network. Once the device has joined the network, its integration into the network may require the user to start the integration process by pressing a button or similar, in order to discover other devices that it can talk to. For example, a device implementing a weigh scale conforming to the ZHC profile tries to find devices containing ZHC aggregation devices (similar to the Continua PHG) to which it could potentially send its measurement data.

The ZigBee pairing process allows for fast and easy association between devices. There are a variety of routing algorithms for data packets to find the correct destination, including neighbour and table-based routing. These approaches result in a high degree of flexibility and stability ensuring that devices in the network stay connected and that network performance remains constant even as it is dynamically changing. ZigBee Health Care offers several way of "pairing" devices.

- End device bind
  - This is a simple push button pair when a button is pressed on 2 devices within a time window and if their services match a "binding" is created
- Service discovery
  - A Health Care device can build a list of Health Care device on the network, for example by listening for new devices to join the network, or by sending a service discovery broadcast to which matching device will respond. The device can now pick which device it would like to communicate with
- Commissioning tool
  - Mandatory primitives in the ZigBee stack allow for a device to query other devices for their services and set up "bindings" and relationships between devices

II.3 ZigBee security

ZigBee security [ZigBee HCP], which is based on a 128-bit AES algorithm, adds to the security model provided by [b-IEEE 802.15.4]. ZigBee's security services include methods for key establishment and transport, device management, and frame protection. Security for Health Care applications is specified as part of the default ZigBee stack profiles, with support for a network key and link keys for point-to-point secure links. In a Health Care network, the aggregator device (often the Continua PHG) will contain a function called the Trust Center. The Trust Center decides whether to allow or disallow new devices into its network. The Trust Center may periodically update and switch to a new Network Key, and controls deployment of link keys. The Trust Center is usually also the network coordinator.

Appendix III Recommendation for use of generic USB drivers

(This appendix does not form an integral part of these guidelines)

It is recommended that Managers for USB PHDC that provide a USB PHDC driver based on a generic USB driver use the following values in the INF file:

Attribute	**INF file element**	**WinUSB value**	**LibUSB value**
Device Class GUID	[Version]/ ClassGUID	{182A3B42-D570-4066-8D13-C72202B40D78}	{EB781AAF-9C70-4523-A5DF-642A87ECA567}
Device Class Text	[Version]/Class [Strings]/ClassName	PHDC	libusb-win32 devices
Interface GUID	[Dev_AddReg]	{B8B610DE-FB41-40A1-A4D6-AB28E87C5F08}	N/A
Device GUID	[Strings]/DeviceGUID	N/A	D0C36FAA-CE6D-4887-A3AA-6FC42D3037E5}

For more information see [b-CHA USB-PHDC].

Bibliography

See [H.810] for a list of non-normative references and publications that contain further background information.