HOW PROTOCOL GATEWAYS GET CONFIGURED

CHARIOT

BY CHIPKIN

You have a Modbus TCP Energy Meter

You have a Modbus TCP Energy Meter

And you have a BACnet/IP Building Management System

You have a Modbus TCP Energy Meter

And you have a BACnet/IP Building Management System

And now ?

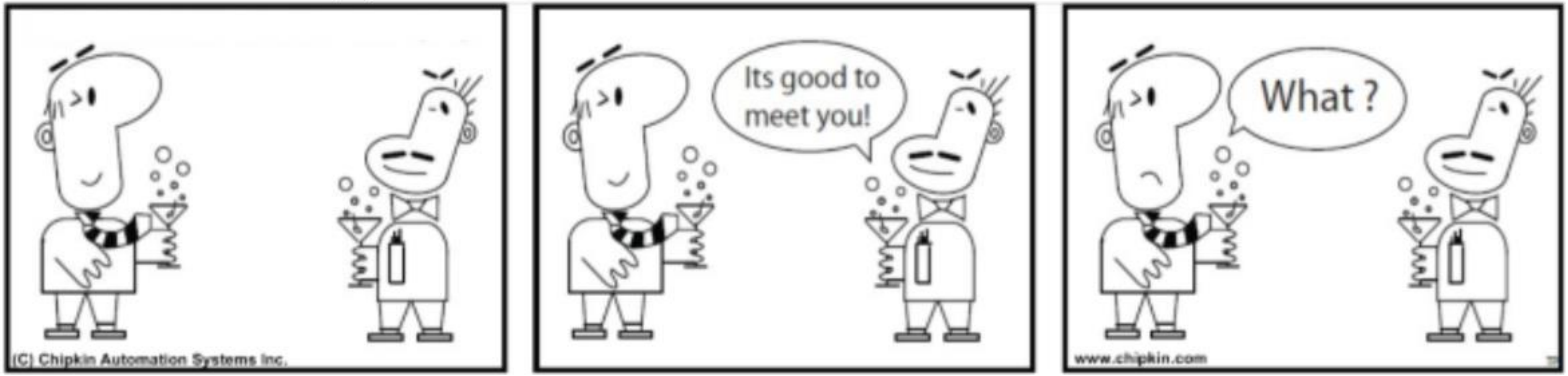
You have a Modbus TCP Energy Meter

And you have a BACnet/IP Building Management System

And now ?

I am reminded of the joke about
Mr Modbus and Mr BACnet

Mr Modbus and Mr BACnet meet at a party ...



You have a Modbus TCP Energy Meter

And you have a BACnet/IP Building Management System

I know about 40001

I know about AO(1)

They don't address data the same way

They don't speak the same language.

CHARIOT
BY CHIPKIN

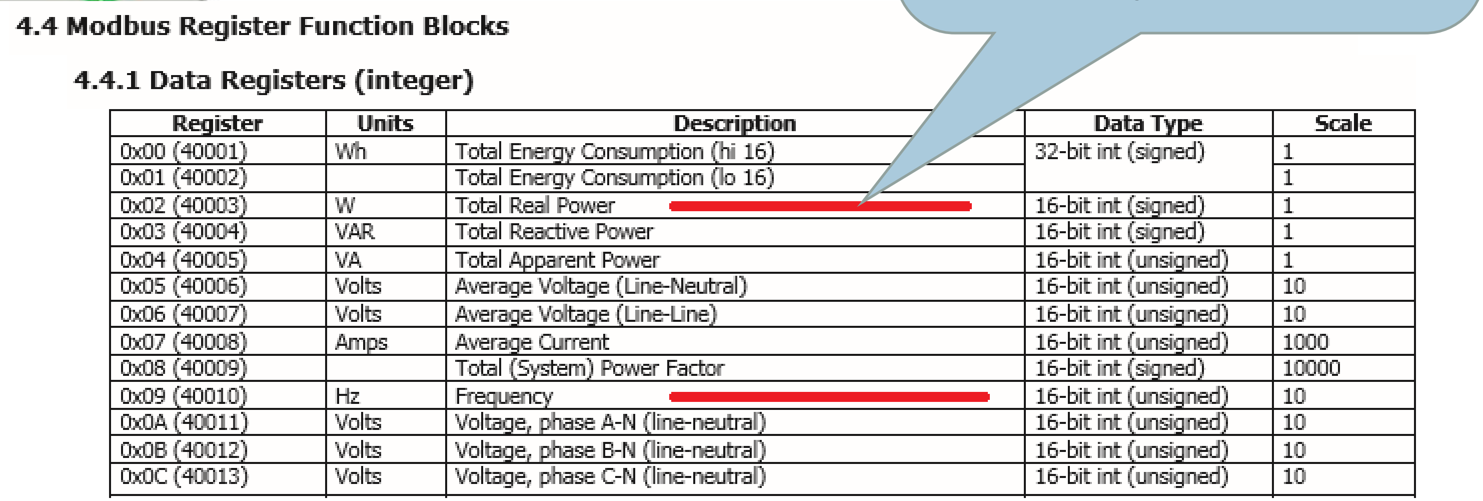
You have a Modbus TCP Energy Meter

And you have a BACnet/IP Building Management System

You need a gateway because the protocols are different. The messages are formatted differently and data is addressed in different ways

CHARIOT
BY CHIPKIN

Lets look at the Modbus Energy Meter



4.4 Modbus Register Function Blocks

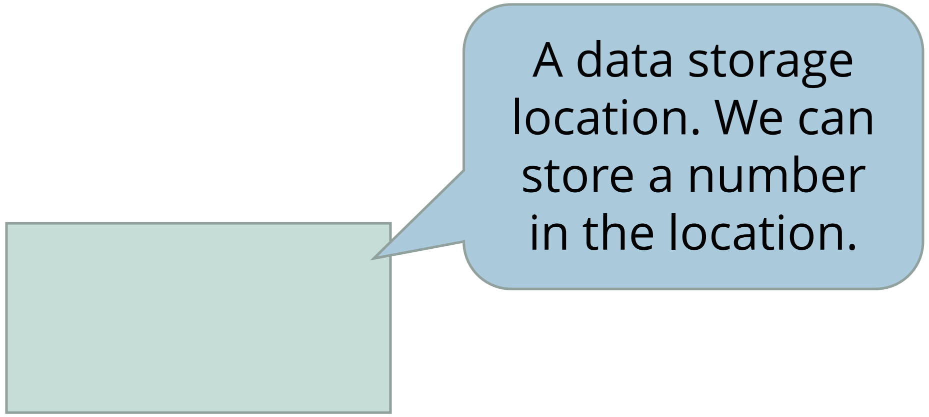
4.4.1 Data Registers (integer)

Register	Units	Description	Data Type	Scale
0x00 (40001)	Wh	Total Energy Consumption (hi 16)	32-bit int (signed)	1
0x01 (40002)		Total Energy Consumption (lo 16)		1
0x02 (40003)	W	Total Real Power	16-bit int (signed)	1
0x03 (40004)	VAR	Total Reactive Power	16-bit int (signed)	1
0x04 (40005)	VA	Total Apparent Power	16-bit int (unsigned)	1
0x05 (40006)	Volts	Average Voltage (Line-Neutral)	16-bit int (unsigned)	10
0x06 (40007)	Volts	Average Voltage (Line-Line)	16-bit int (unsigned)	10
0x07 (40008)	Amps	Average Current	16-bit int (unsigned)	1000
0x08 (40009)		Total (System) Power Factor	16-bit int (signed)	10000
0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10
0x0A (40011)	Volts	Voltage, phase A-N (line-neutral)	16-bit int (unsigned)	10
0x0B (40012)	Volts	Voltage, phase B-N (line-neutral)	16-bit int (unsigned)	10
0x0C (40013)	Volts	Voltage, phase C-N (line-neutral)	16-bit int (unsigned)	10

Your job is to configure a gateway to deliver the selected Modbus Data to the BACnet system

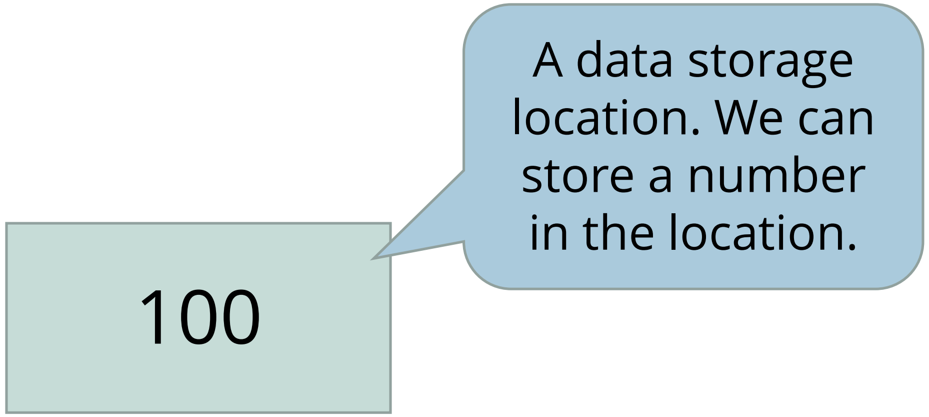
This is how to configure a gateway to integrate the Modbus Energy data into a BACnet Building System

First we need to define a cache to store some data



A data storage location. We can store a number in the location.

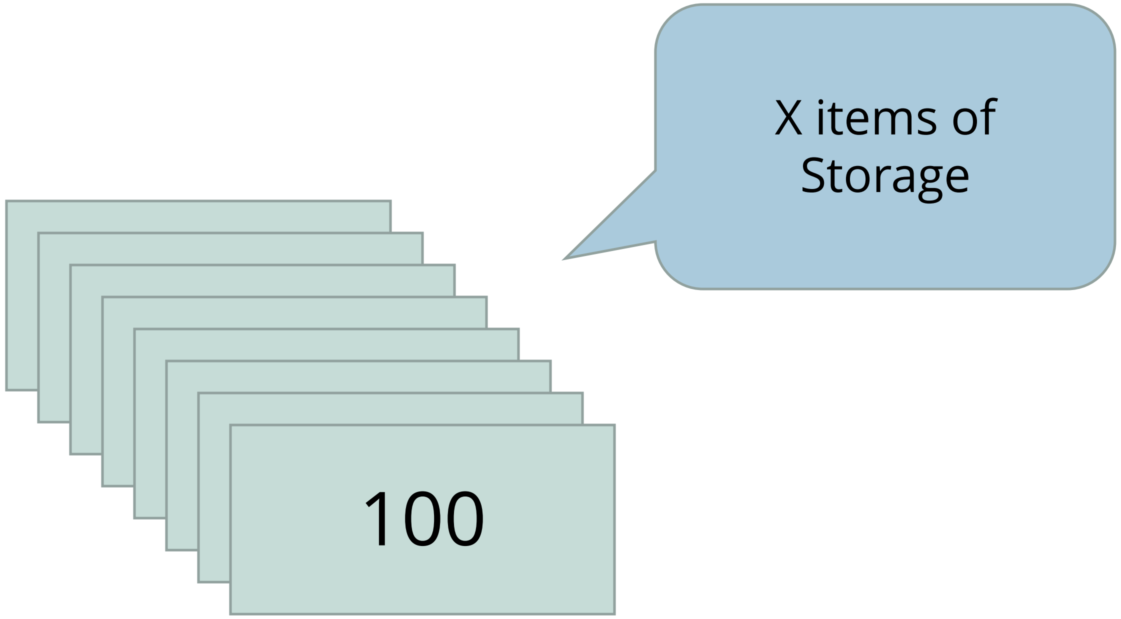
First we need to define a cache to store some data



100

A data storage location. We can store a number in the location.

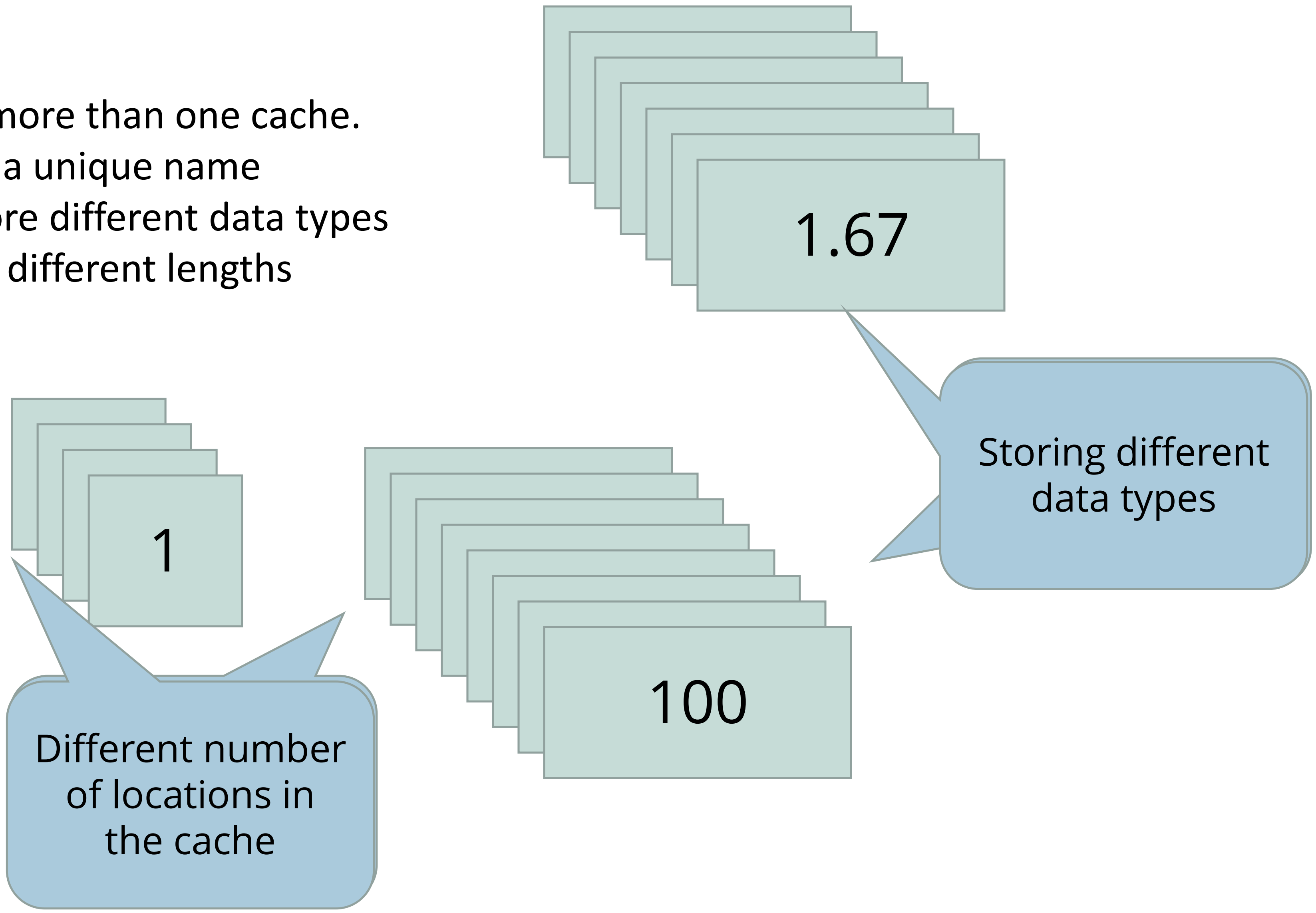
A data storage cache could be an array.



X items of Storage

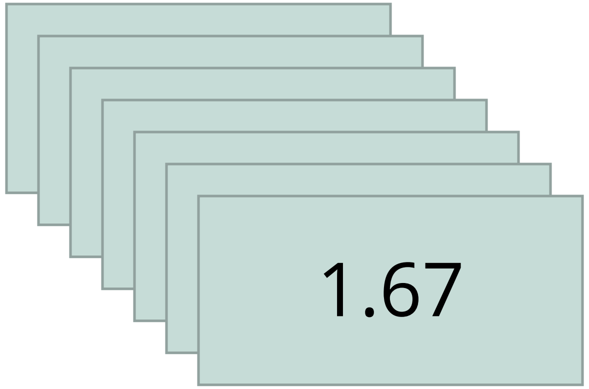
100

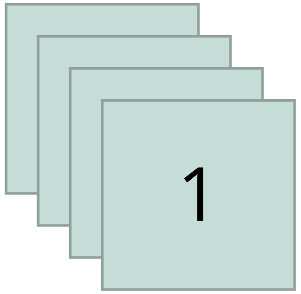
There can be more than one cache.
Each will have a unique name
Each might store different data types
They might be different lengths

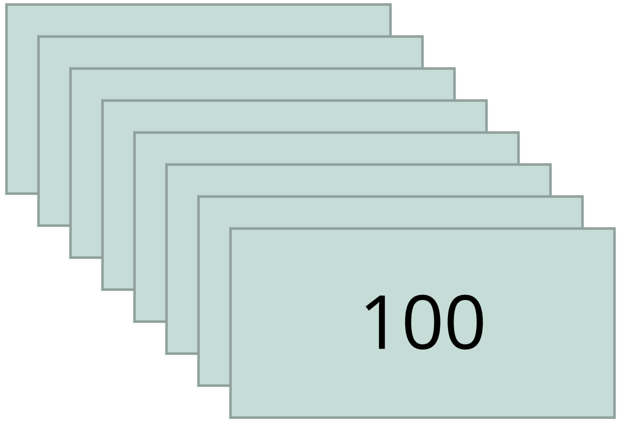

1.67
Storing different data types
1
100
Different number of locations in the cache

3 Key Properties

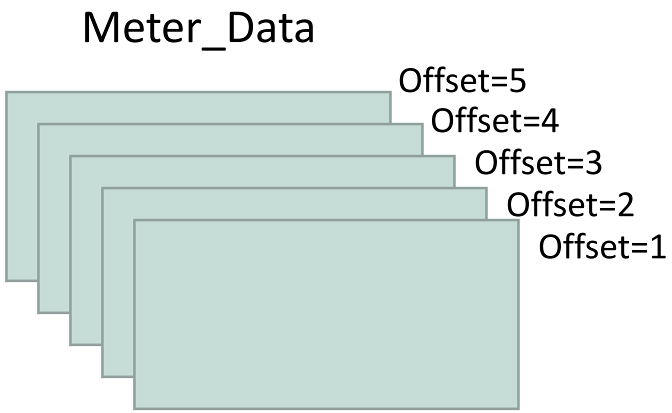
1. What kind of number can be stored
2. How many items in the cache
3. Cache name



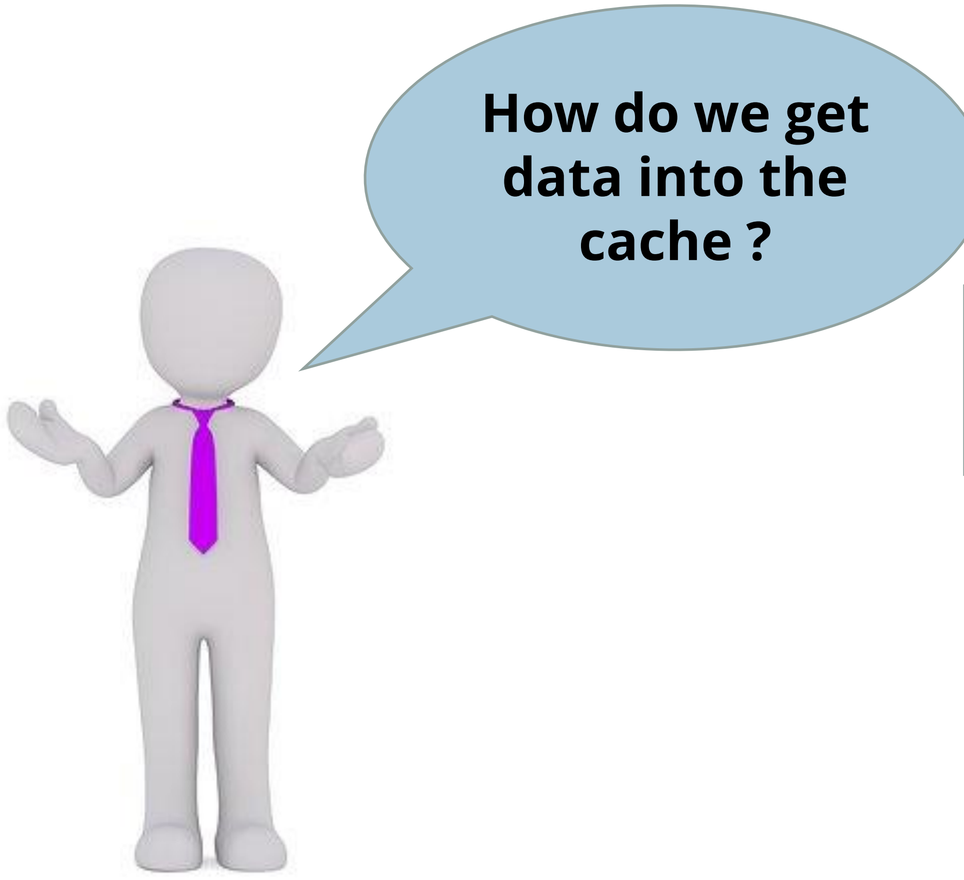




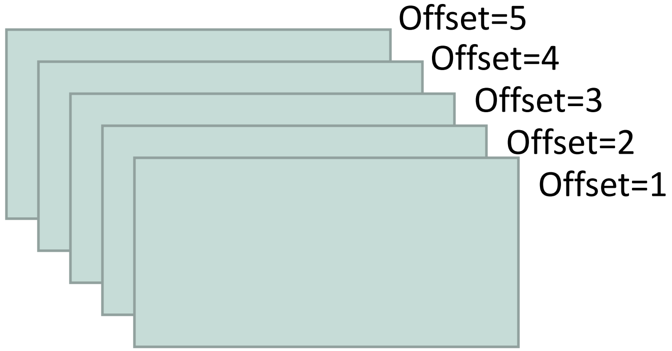
Example



Data Type = Float
Number of Items = 5

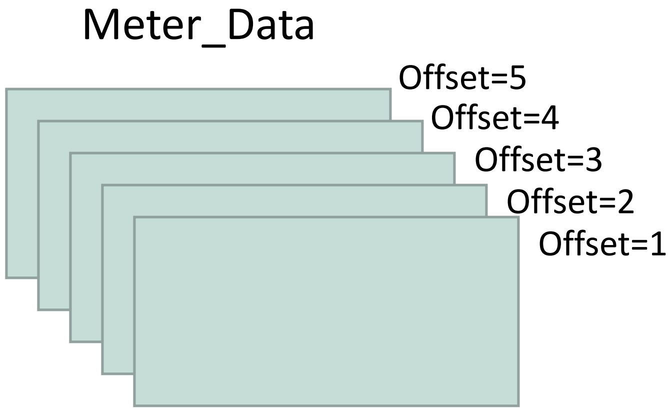


Meter_Data



Offset=5
Offset=4
Offset=3
Offset=2
Offset=1

Data Type = Float
Number of Items = 5



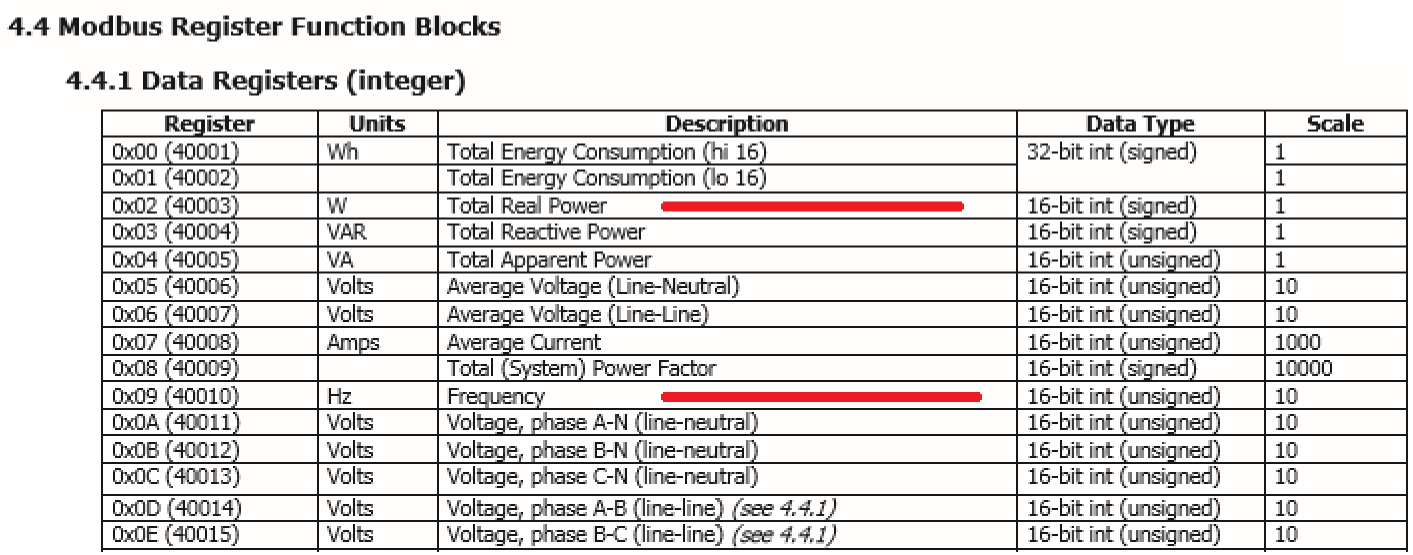
Meter_Data

Offset=5
Offset=4
Offset=3
Offset=2
Offset=1

Data Type = Float
Number of Items = 5

Lets look at The Modbus Points List

This customer want to convert 2 points to BACnet. They are marked in Red



4.4 Modbus Register Function Blocks

4.4.1 Data Registers (integer)

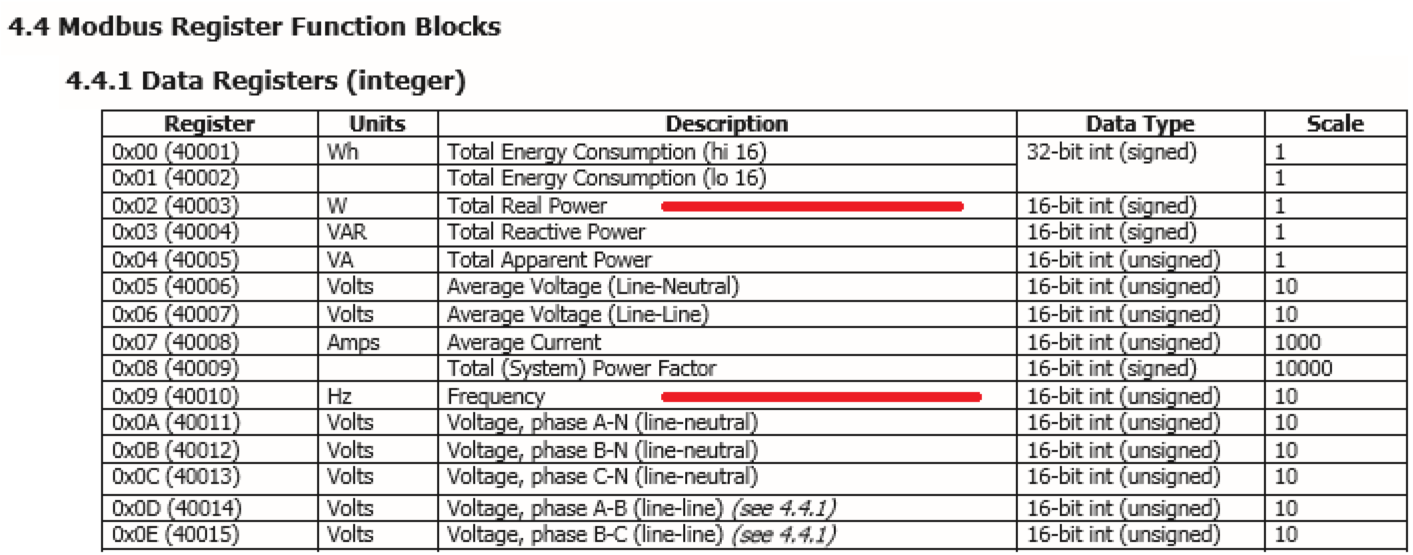
Register	Units	Description	Data Type	Scale
0x00 (40001)	Wh	Total Energy Consumption (hi 16)	32-bit int (signed)	1
0x01 (40002)		Total Energy Consumption (lo 16)		1
0x02 (40003)	W	Total Real Power	16-bit int (signed)	1
0x03 (40004)	VAR	Total Reactive Power	16-bit int (signed)	1
0x04 (40005)	VA	Total Apparent Power	16-bit int (unsigned)	1
0x05 (40006)	Volts	Average Voltage (Line-Neutral)	16-bit int (unsigned)	10
0x06 (40007)	Volts	Average Voltage (Line-Line)	16-bit int (unsigned)	10
0x07 (40008)	Amps	Average Current	16-bit int (unsigned)	1000
0x08 (40009)		Total (System) Power Factor	16-bit int (signed)	10000
0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10
0x0A (40011)	Volts	Voltage, phase A-N (line-neutral)	16-bit int (unsigned)	10
0x0B (40012)	Volts	Voltage, phase B-N (line-neutral)	16-bit int (unsigned)	10
0x0C (40013)	Volts	Voltage, phase C-N (line-neutral)	16-bit int (unsigned)	10
0x0D (40014)	Volts	Voltage, phase A-B (line-line) *(see 4.4.1)*	16-bit int (unsigned)	10
0x0E (40015)	Volts	Voltage, phase B-C (line-line) *(see 4.4.1)*	16-bit int (unsigned)	10

CHARIOT BY CHIPKIN

Lets look at The Modbus Points List

This customer want to convert 2 points to BACnet. They are marked in Red

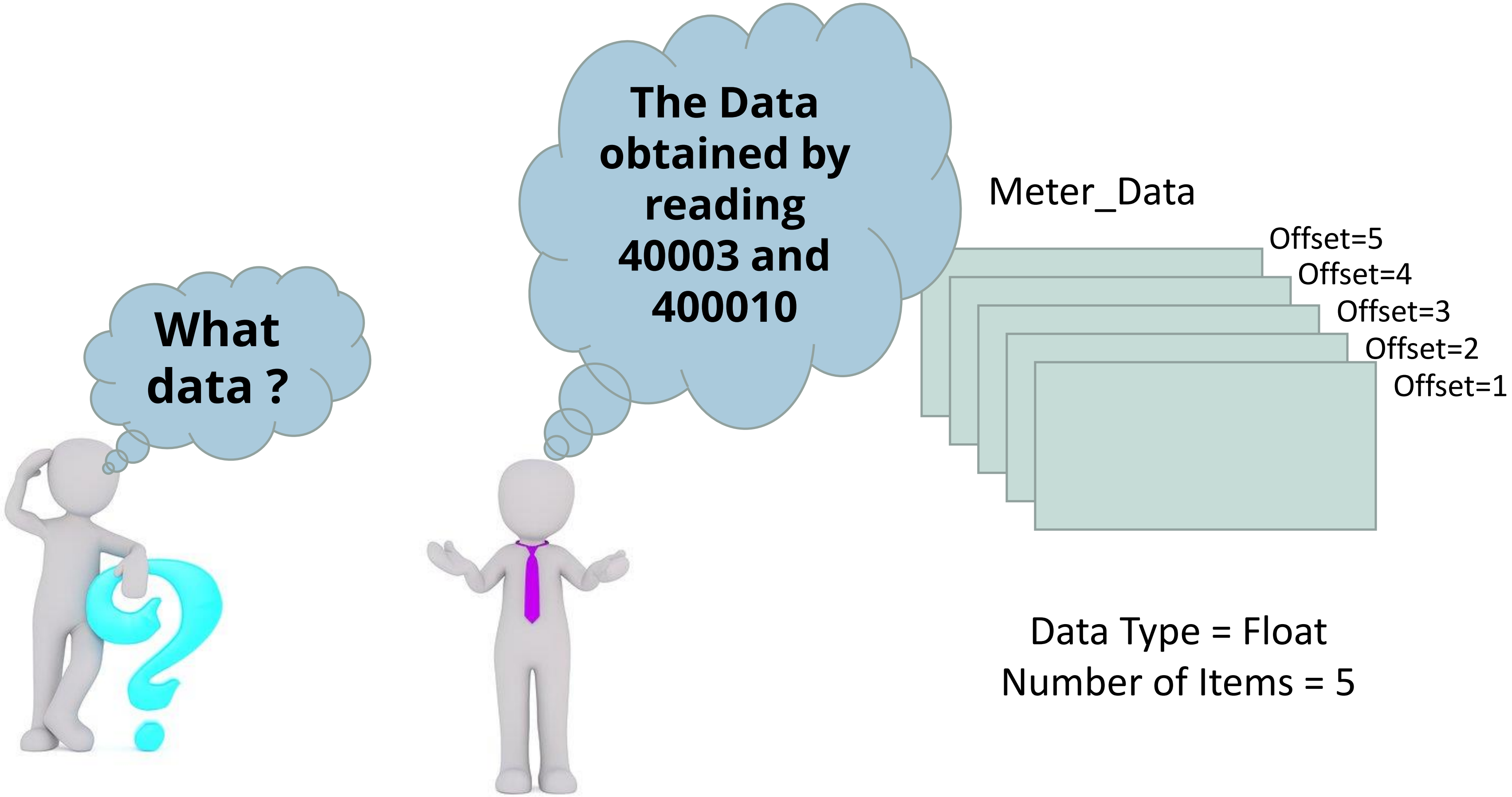
40003 Total Real Power and 40010 Frequency



4.4 Modbus Register Function Blocks

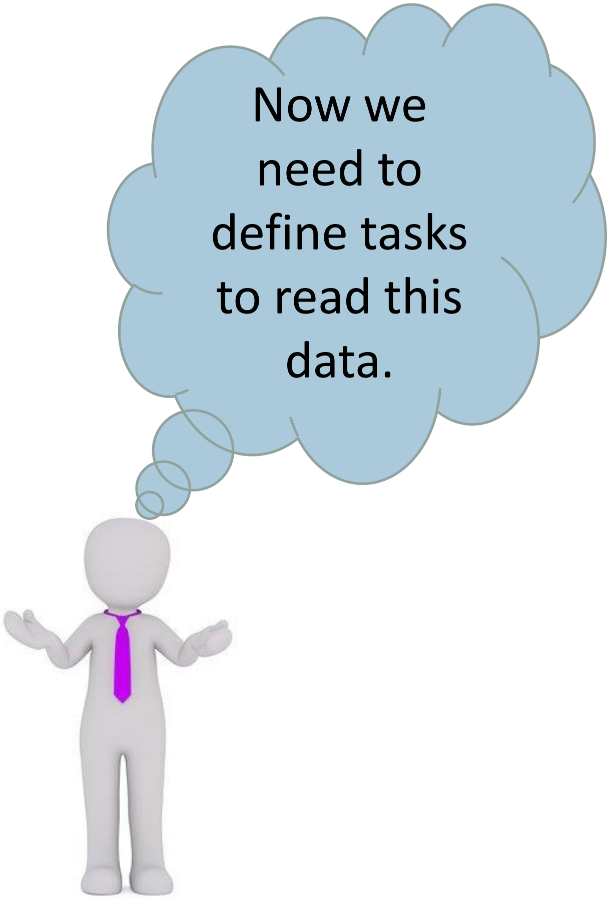
4.4.1 Data Registers (integer)

Register	Units	Description	Data Type	Scale
0x00 (40001)	Wh	Total Energy Consumption (hi 16)	32-bit int (signed)	1
0x01 (40002)		Total Energy Consumption (lo 16)		1
0x02 (40003)	W	Total Real Power	16-bit int (signed)	1
0x03 (40004)	VAR	Total Reactive Power	16-bit int (signed)	1
0x04 (40005)	VA	Total Apparent Power	16-bit int (unsigned)	1
0x05 (40006)	Volts	Average Voltage (Line-Neutral)	16-bit int (unsigned)	10
0x06 (40007)	Volts	Average Voltage (Line-Line)	16-bit int (unsigned)	10
0x07 (40008)	Amps	Average Current	16-bit int (unsigned)	1000
0x08 (40009)		Total (System) Power Factor	16-bit int (signed)	10000
0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10
0x0A (40011)	Volts	Voltage, phase A-N (line-neutral)	16-bit int (unsigned)	10
0x0B (40012)	Volts	Voltage, phase B-N (line-neutral)	16-bit int (unsigned)	10
0x0C (40013)	Volts	Voltage, phase C-N (line-neutral)	16-bit int (unsigned)	10
0x0D (40014)	Volts	Voltage, phase A-B (line-line) *(see 4.4.1)*	16-bit int (unsigned)	10
0x0E (40015)	Volts	Voltage, phase B-C (line-line) *(see 4.4.1)*	16-bit int (unsigned)	10



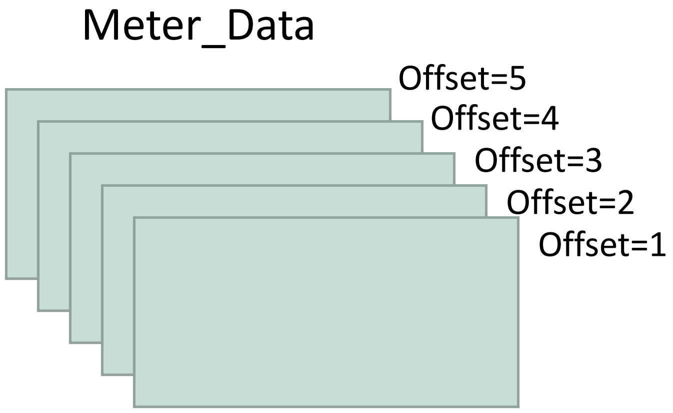
Data Type = Float
Number of Items = 5

Now we need to define tasks to read this data.



Task 1:
Read 40003 and
store the data in
Cache offset=1

Task 2:
Read 40010 and
store the data in
Cache offset=2


Meter_Data

Offset=5
Offset=4
Offset=3
Offset=2
Offset=1

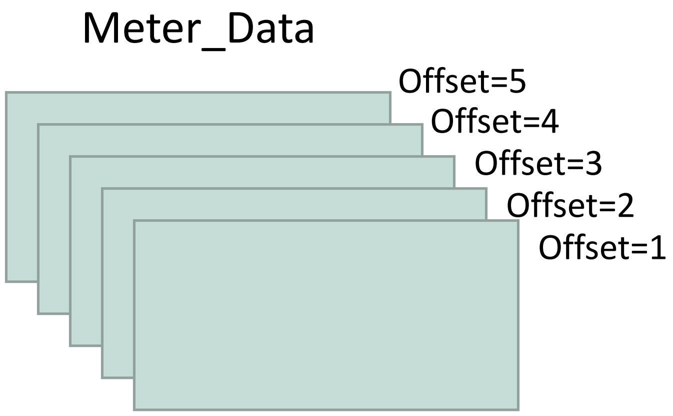
Data Type = Float
Number of Items = 5

Now we need to define tasks to read this data.

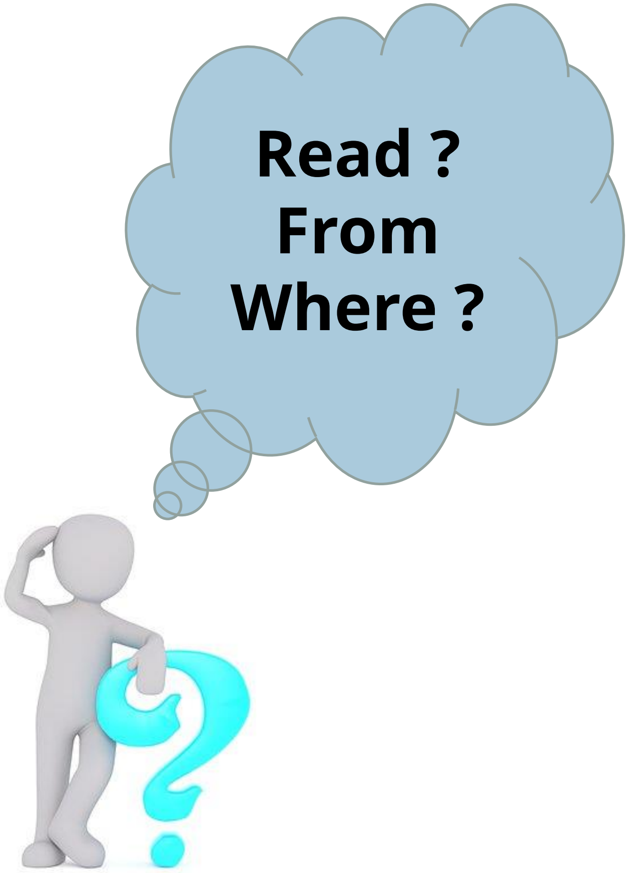
Task 1:
Read 40003 and store the data in Cache offset=1

Task 2:
Read 40010 and store the data in Cache offset=2

Where you store the data in the cache is an arbitrary choice.

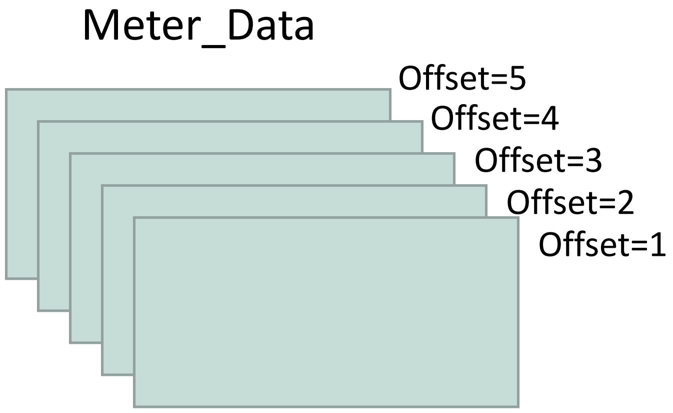


Data Type = Float
Number of Items = 5



Task 1:
Read 40003 and store the data in Cache offset=1

Task 2:
Read 40010 and store the data in Cache offset=2


Meter_Data

Offset=5

Offset=4

Offset=3

Offset=2

Offset=1

Data Type = Float

Number of Items = 5

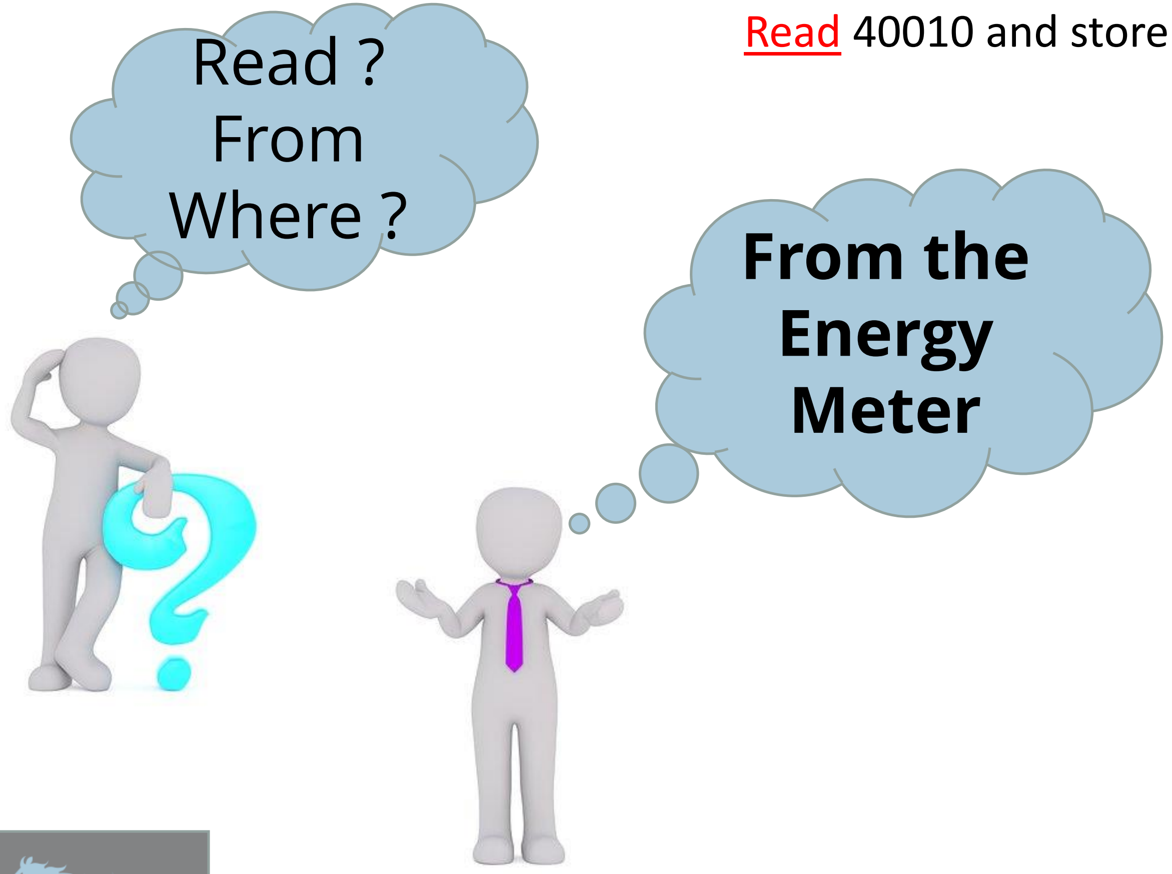
CHARIOT
BY CHIPKIN

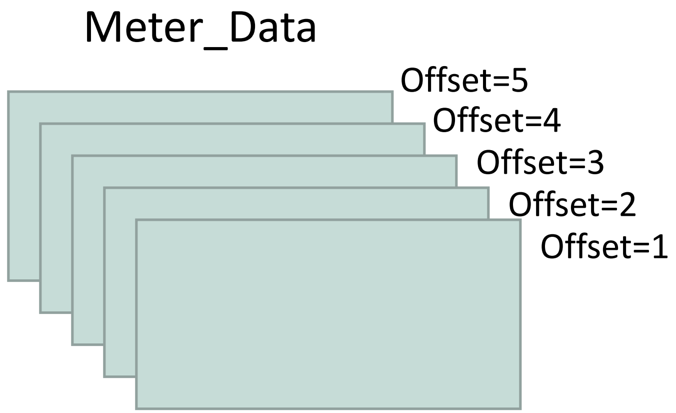
Task 1:

Read 40003 and store the data in Cache offset=1

Task 2:

Read 40010 and store the data in Cache offset=2





Data Type = Float

Number of Items = 5

What is the Meter's
IP Address ?

What is the Meter's
Modbus NodeID ?

What is the Meter’s
IP Address ?
= 192.168.1.168

What is the Meter’s
Modbus NodeID ?
= 1

And therefore …

Task 1:
Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1

Task 2:
Read 40010 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=2

And therefore …

Task 1:
Read 40003 From Node=1
data in Cache called Meter_

How Often ?
As often as possible, once, every x seconds ?

Task 2:
Read 40010 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=2

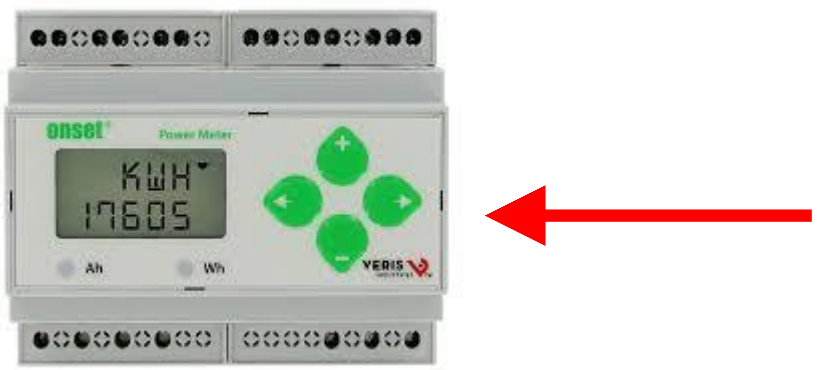
And therefore …

Task 1:
Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1. Do this task continuously

Task 2:
Read 40010 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=2. Do this task continuously

Task 1:

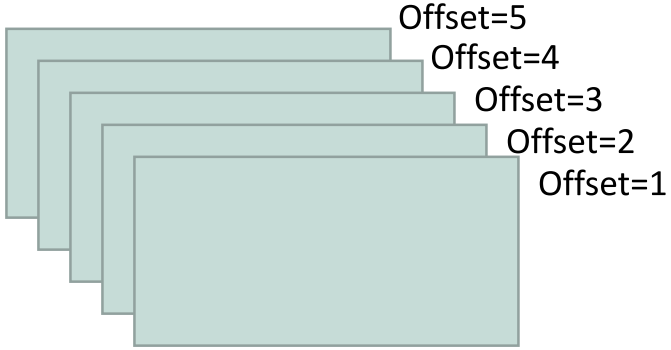
Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1. Do this task continuously



Energy meter is Modbus Server

Gateway as Modbus Master – Sends poll to read data

Meter_Data



Data Type = Float
Number of Items = 5

Task 1:
Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1. Do this task continuously

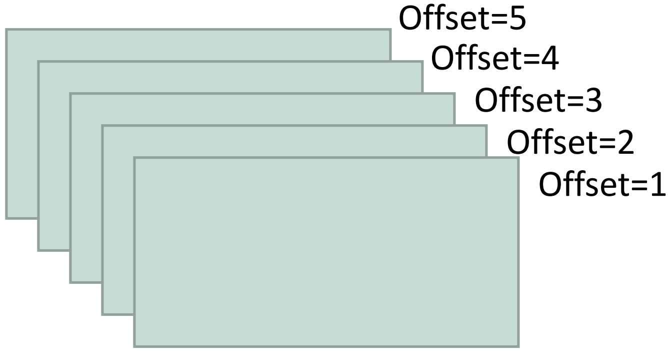
Energy meter is Modbus Server

Gateway as Modbus Master – Sends poll to read data

Responds to poll with data payload

ValueOf(40003)=100 (example)

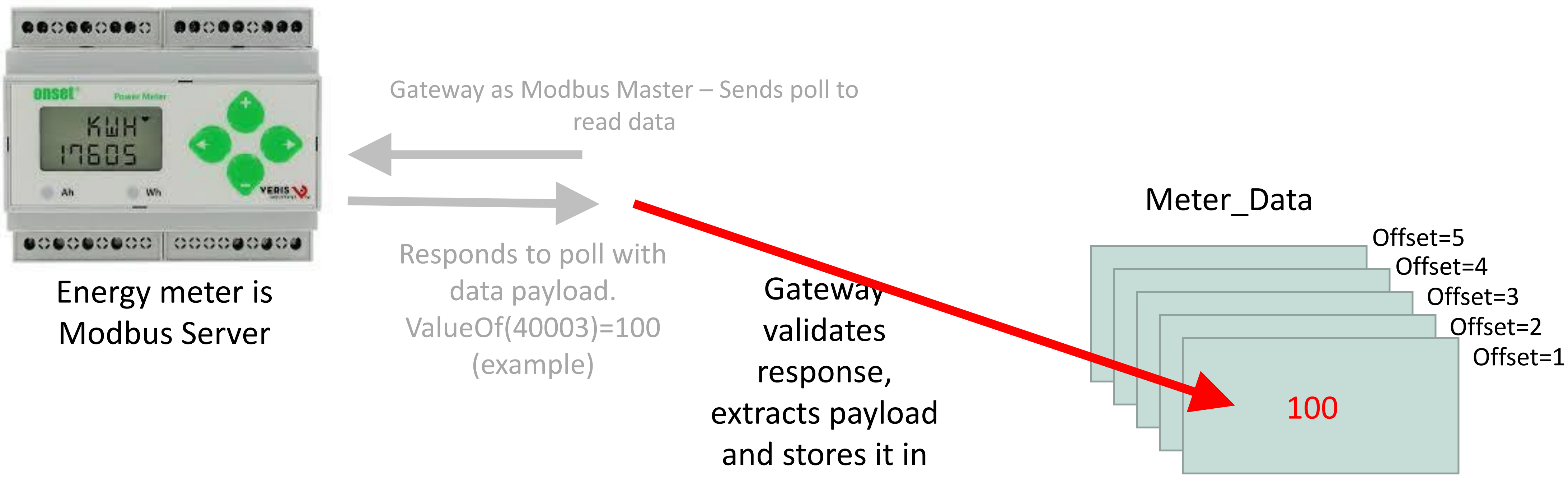
Meter_Data



Data Type = Float
Number of Items = 5

Task 1:

Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1. Do this task continuously



Gateway as Modbus Master – Sends poll to read data

Responds to poll with data payload. ValueOf(40003)=100 (example)

Energy meter is Modbus Server

Gateway validates response, extracts payload and stores it in the cache at offset=1

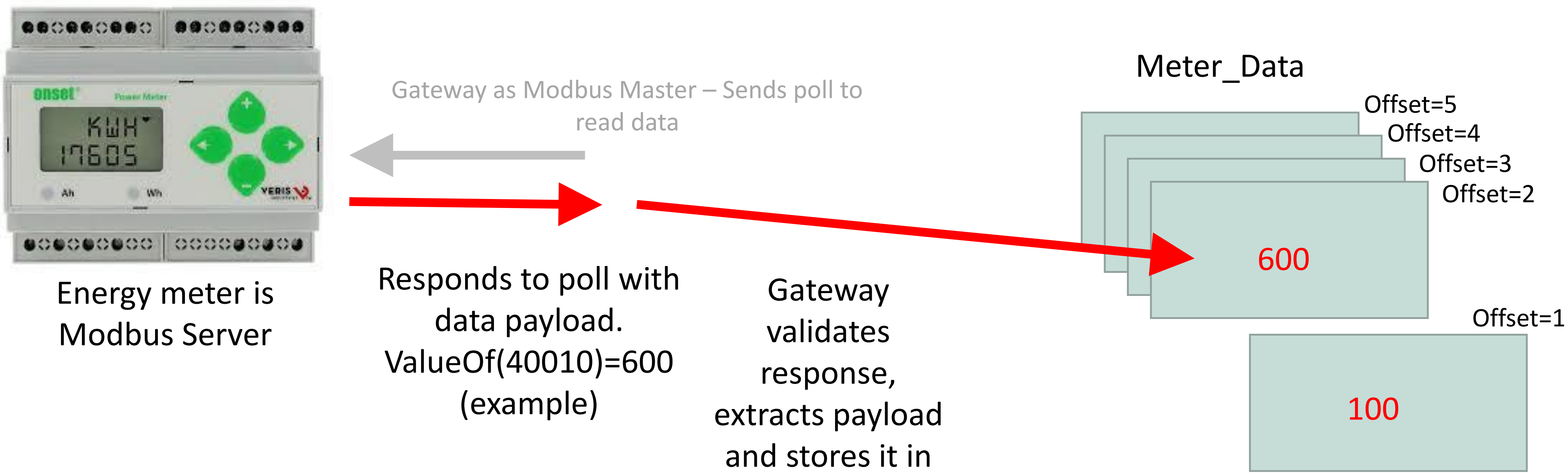
Meter_Data

Data Type = Float
Number of Items = 5

Its similar for task 2

Task 2:

Read 40010 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=2. Do this task continuously



Gateway as Modbus Master – Sends poll to read data

Energy meter is Modbus Server

Responds to poll with data payload. ValueOf(40010)=600 (example)

Gateway validates response, extracts payload and stores it in the cache at offset=2

Meter_Data

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

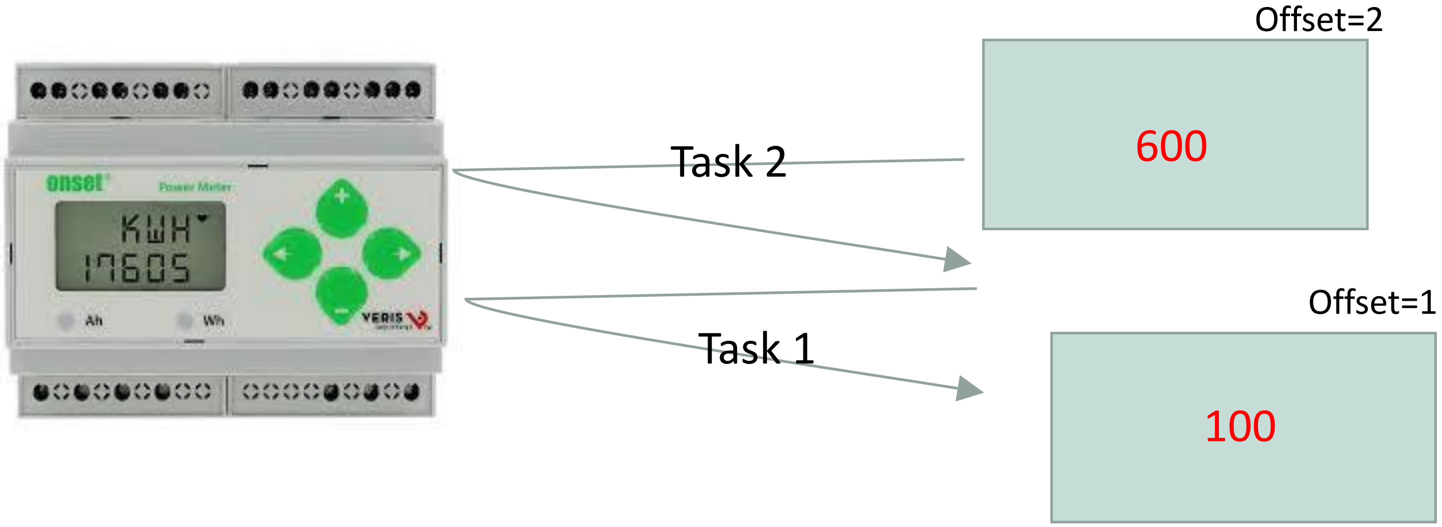
Data Type = Float
Number of Items = 5

Task 1:

Read 40003 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=1. Do this task continuously

Task 2:

Read 40010 From Node=1 at IP=192.168.1.168 and store the data in Cache called Meter_Data at offset=2. Do this task continuously



Offset=2

600

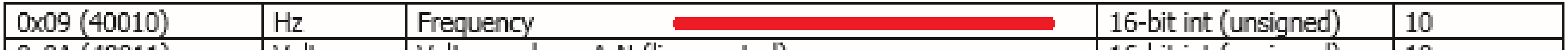
Task 2

Offset=1

Task 1

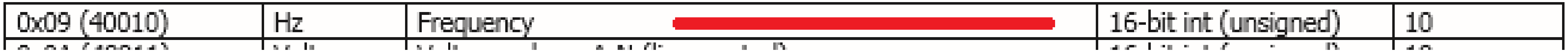
100

That is almost the end of configuring the client (Modbus) side of the gateway



0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10

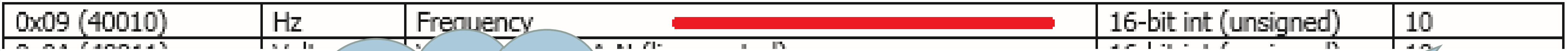
We see the scaling factor is 10.
Meaning if we read a value of 600 in register 40010 then we must divide by 10 to get the frequency of 60



0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10

Therefore we need to scale the one value before serving it on BACnet.

We see the scaling factor is 10.
Meaning if we read a value of 600 in register 40010 then we must divide by 10 to get the frequency of 60



0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10

Therefore we need to scale the one value before serving it on BACnet.

We see the scaling factor is 10.
Meaning if we read a value of 600 in register 40010 then we must divide by 10 to get the frequency of 60

3 Approaches

1. Apply scaling on the read task so it stores the scaled vaslue.
2. Apply scaling when we serve the value (extract it from the cache)
3. Create a task to take the value int the cache , scale it and then store it in a new location in the cache.

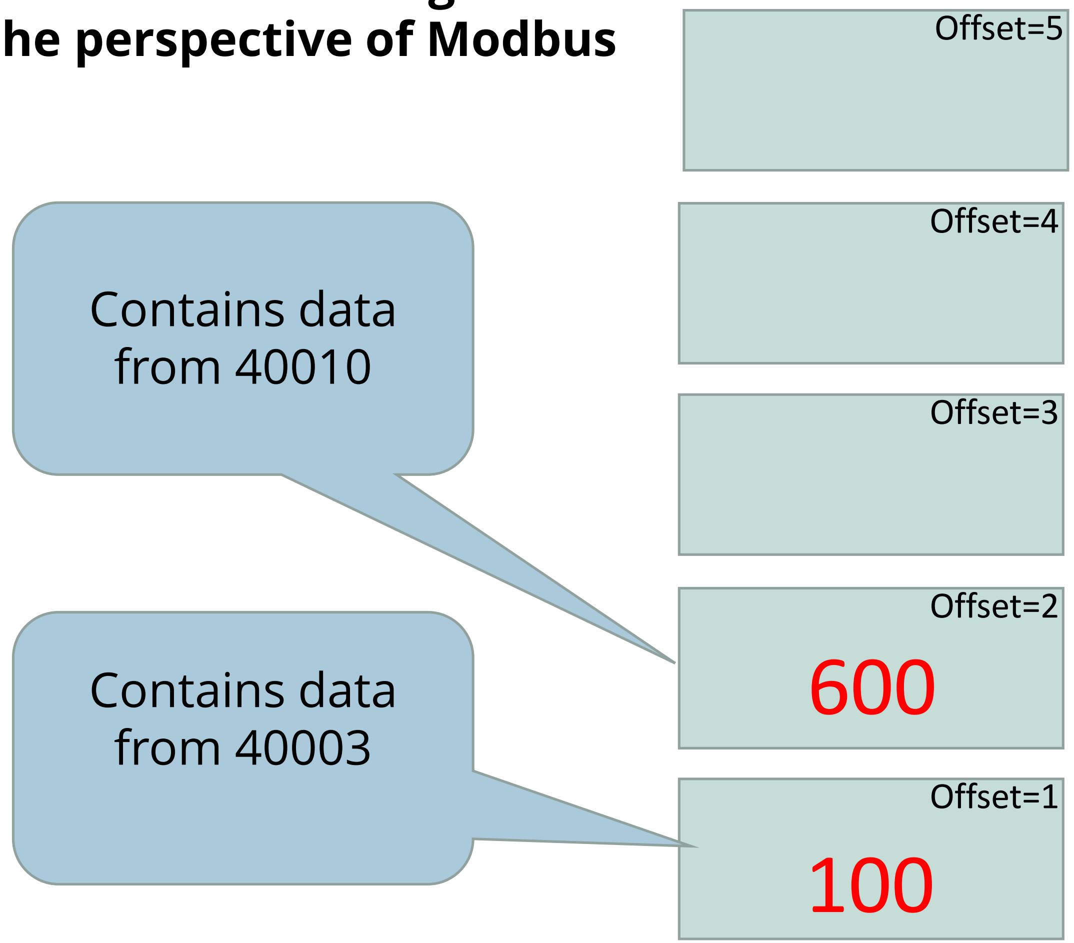
CHARIOT
BY CHIPKIN

Now we must configure the BACnet side of the gateway

Now we must configure the BACnet side of the gateway

- Did the customer constrain us ?
- Ie did they tell us to use particular instance numbers, ….
- Probably not. – Lets assume they did not

Lets Look at the cache again from the perspective of Modbus



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

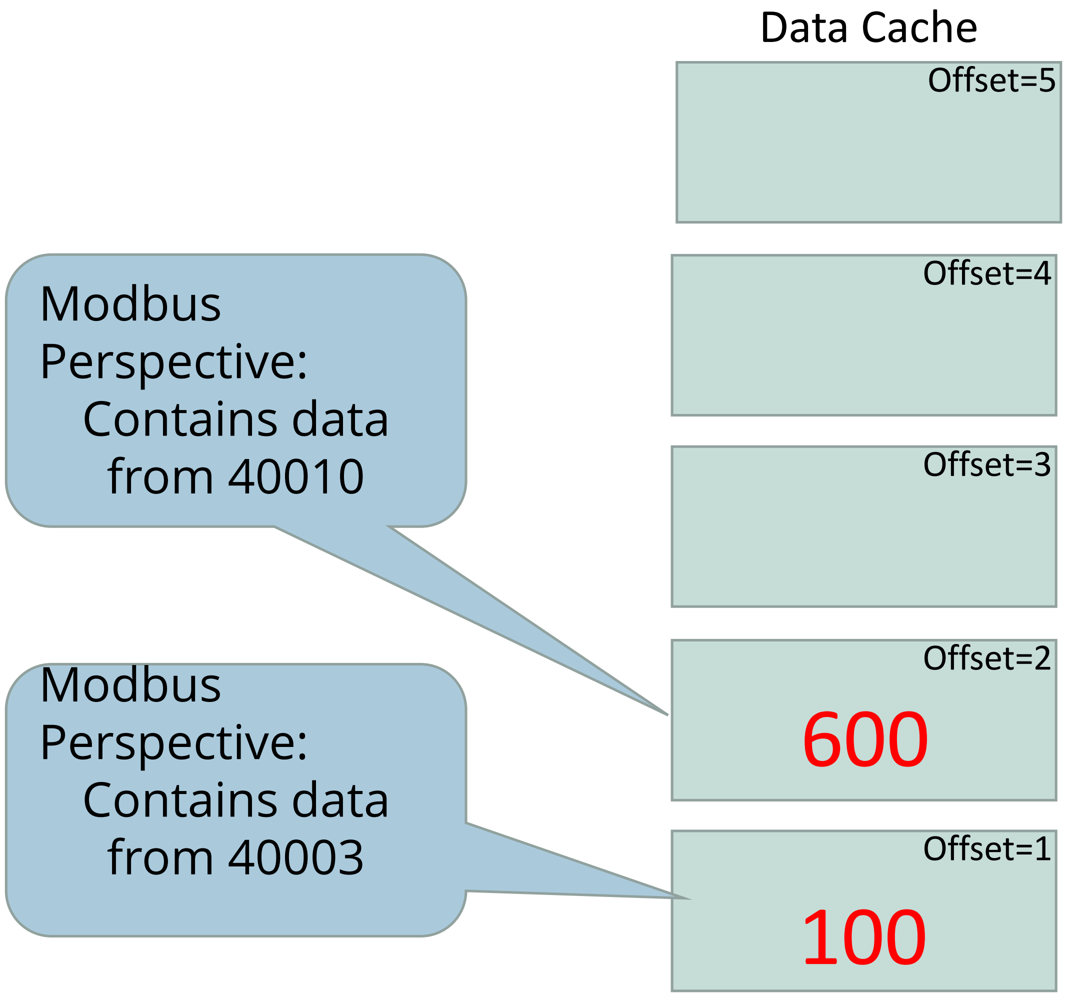
100

Contains data from 40010

Contains data from 40003

Data Type = Float
Number of Items = 5

Lets Look at the cache again from the perspective of BACnet



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

Data Type = Float
Number of Items = 5

Lets Look at the cache again from the perspective of BACnet

Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

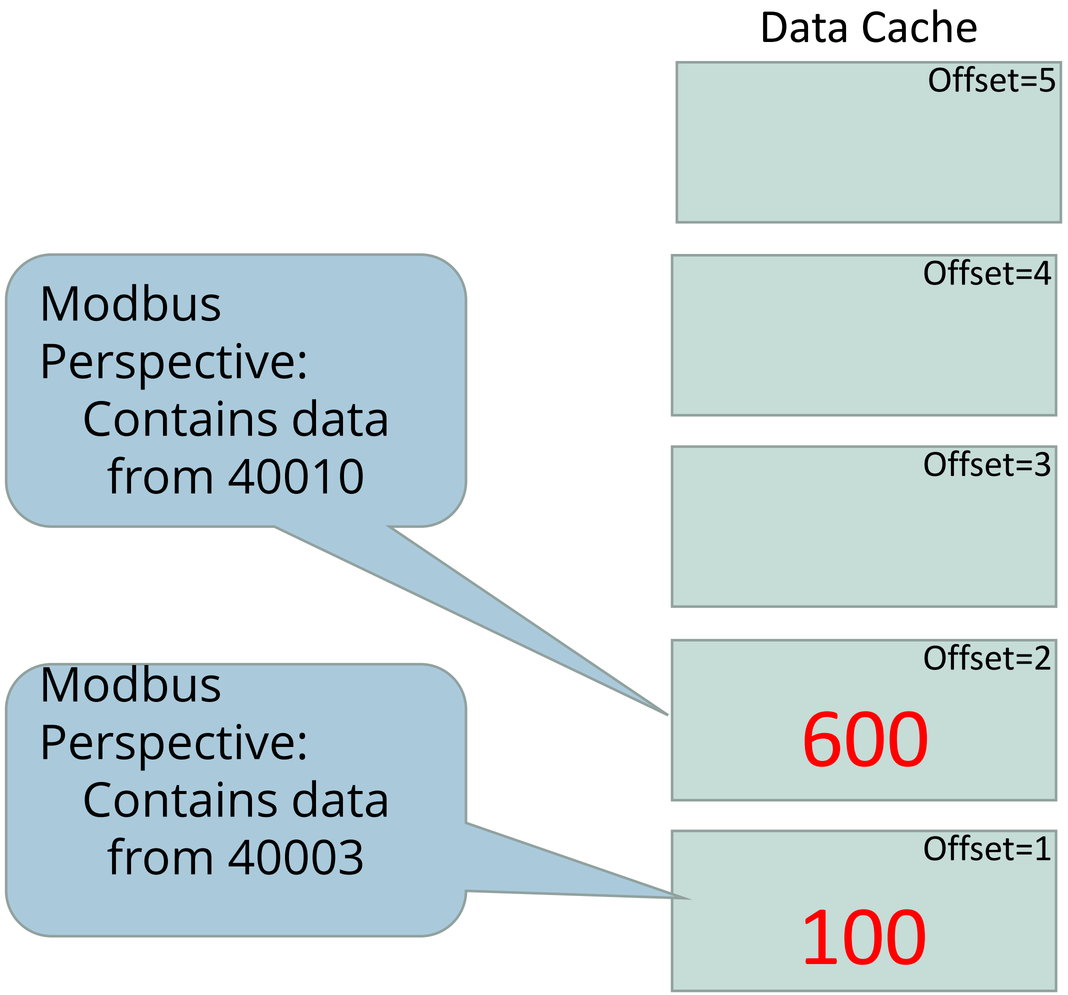
Data Type = Float
Number of Items = 5

Modbus Perspective:
Contains data from 40010

Modbus Perspective:
Contains data from 40003

To get at this data the Building system must send BACnet requests to the gateway

Lets Look at the cache again from the perspective of BACnet



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

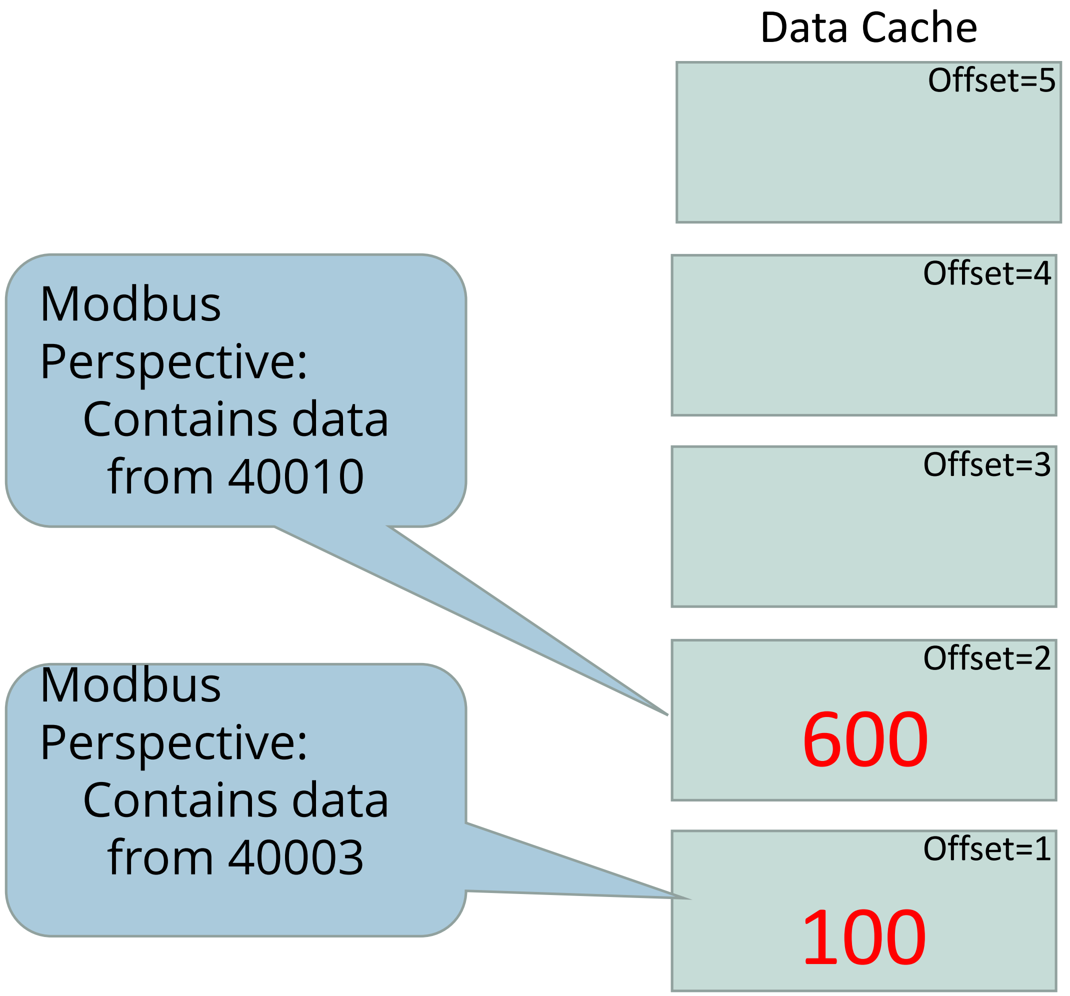
Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

Data Type = Float
Number of Items = 5

To get at this data the Building system must send BACnet requests to the gateway

??? But the BMS doesn’t know where to send messages to.



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

Data Type = Float
Number of Items = 5

This means the BMS needs to know the IP address of the gateway
And
The BACnet device instance number of the gateway

Since the customer did not constrain us (provide direction) we will assign arbitrary values (Might need to change when this gets to site)

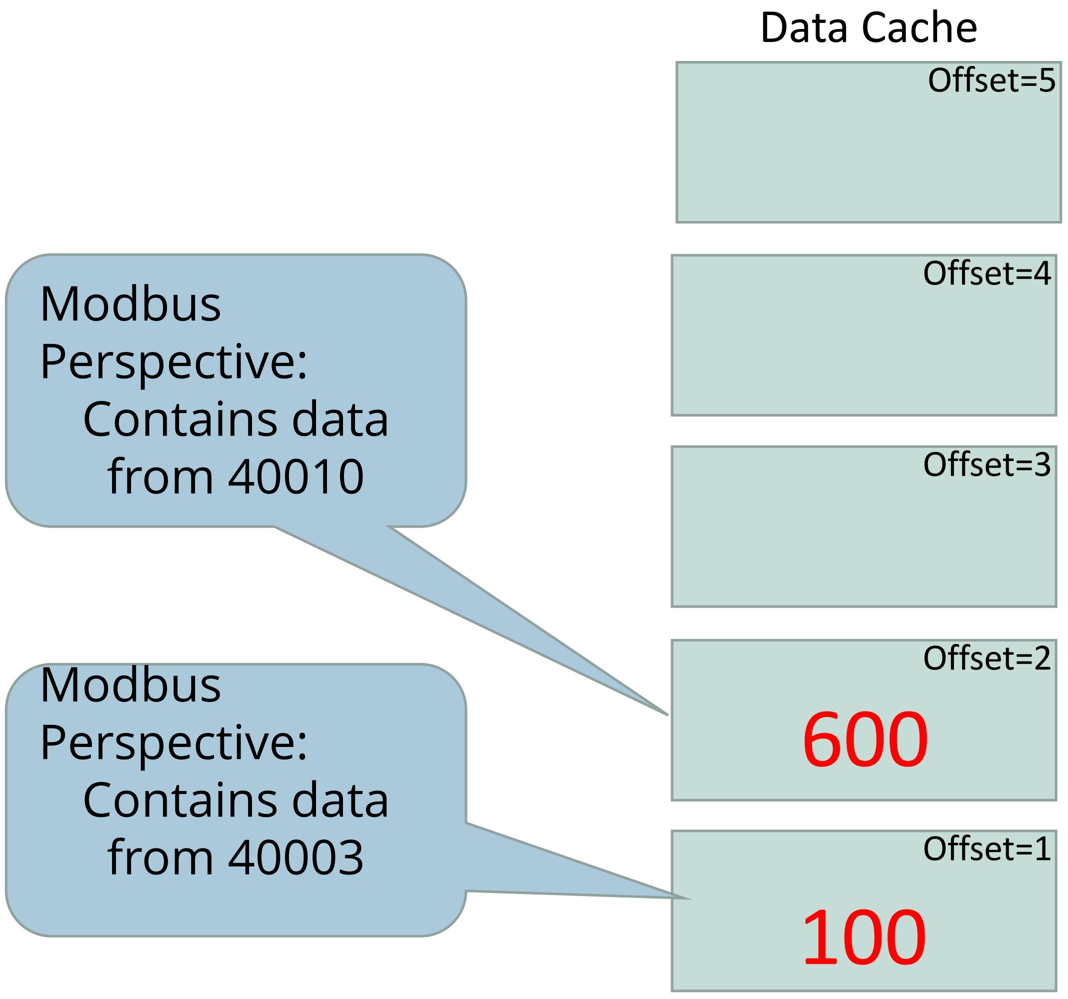
<u>For Example</u>

Gateway IP Address = 10.10.122.10

Should be on the same subnet as the BMS

Device Instance Number = 389001

You can use any number. BACnet vendors have ID's. Chipkin ID=389 therefore it makes sense to number our device 389001



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Modbus Perspective:
Contains data from 40010

Modbus Perspective:
Contains data from 40003

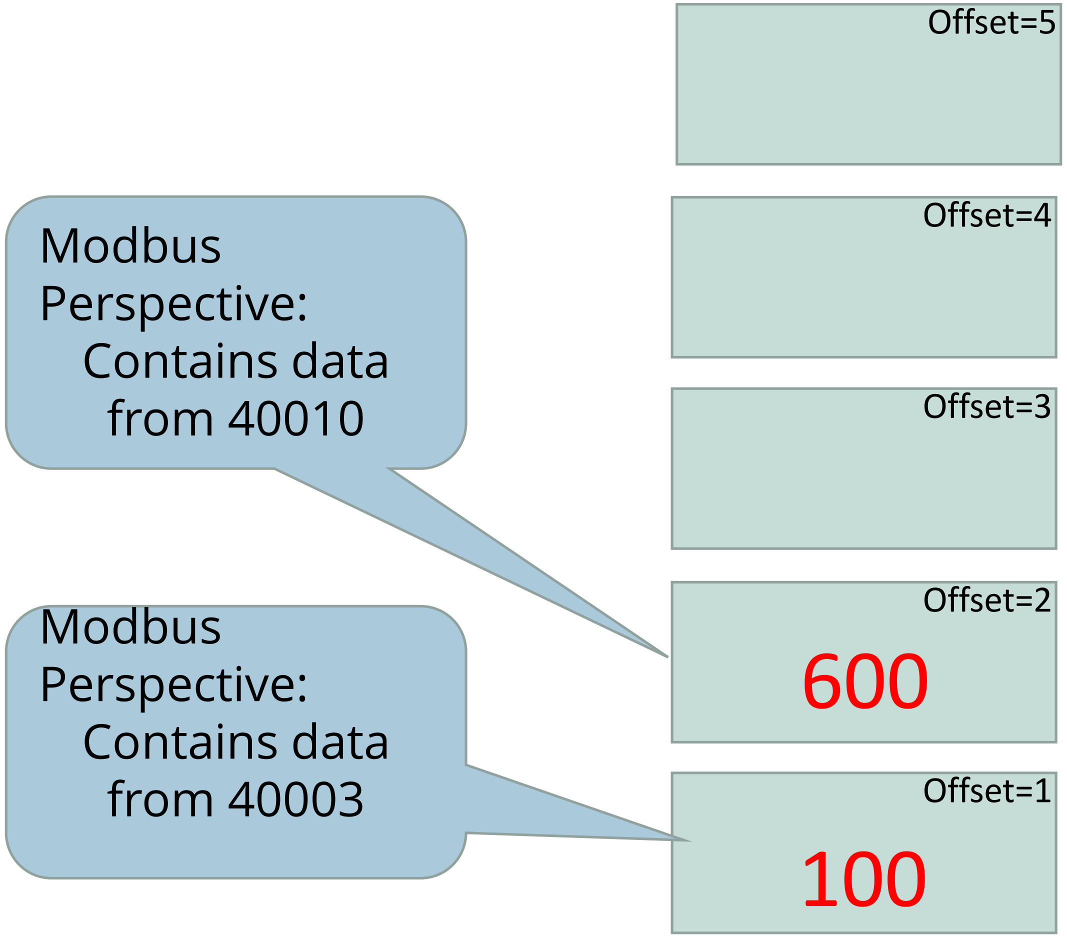
Data Type = Float
Number of Items = 5

Lets Look at the cache again
from the perspective of
BACnet

Now the BMS knows where to
send BACnet messages to.

Gateway IP Address = 10.10.122.10
Device Instance Number = 389001

Data Cache



Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

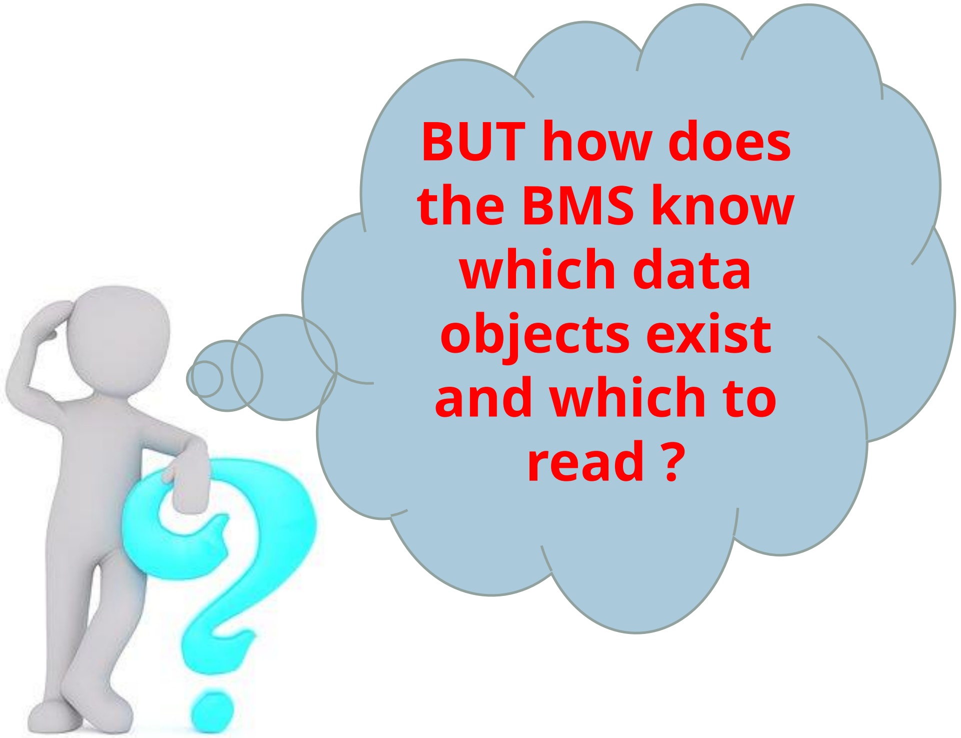
Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

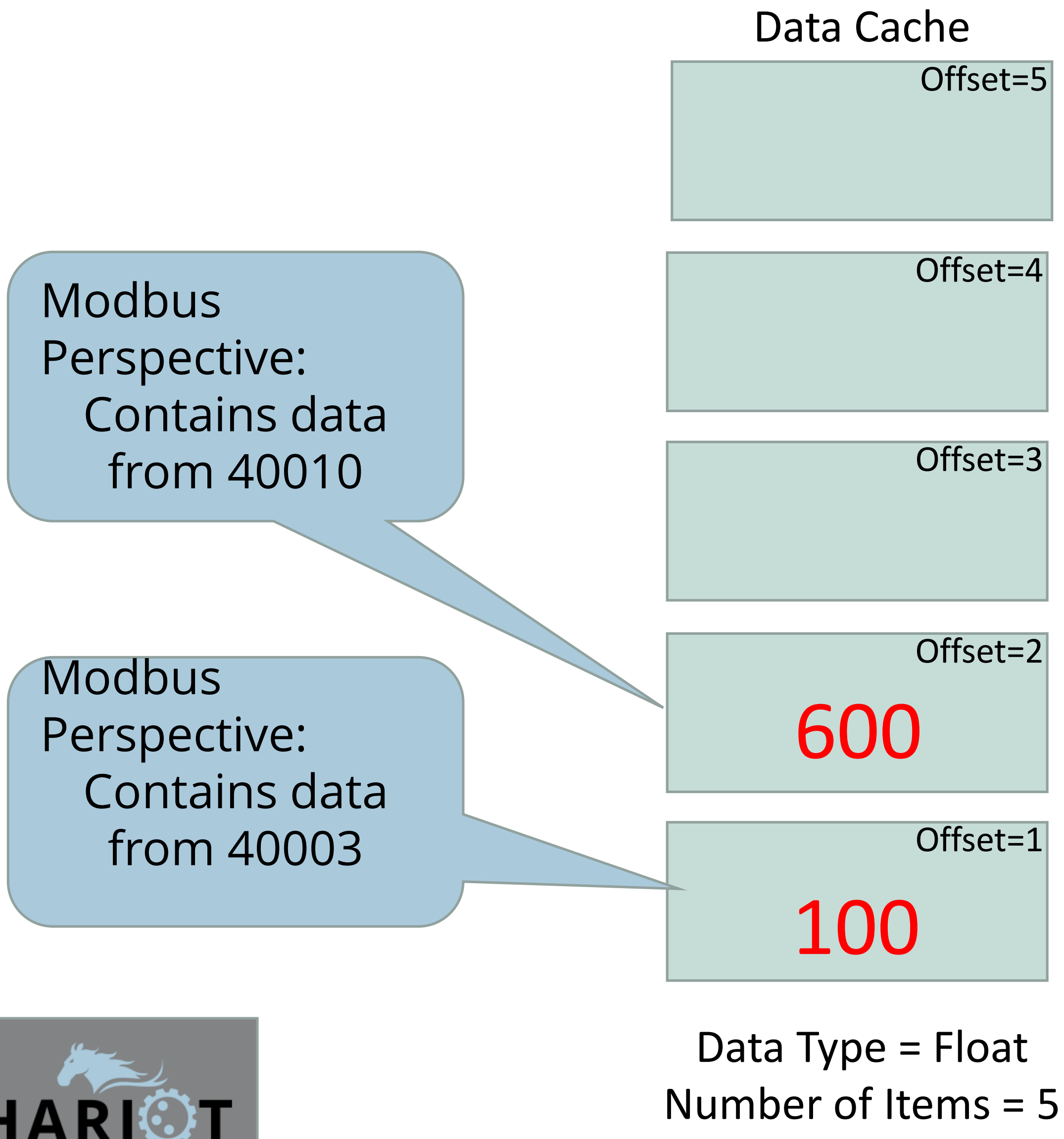
Data Type = Float
Number of Items = 5

From the perspective of BACnet

Gateway IP Address = 10.10.122.10
Device Instance Number = 389001



BUT how does the BMS know which data objects exist and which to read ?



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

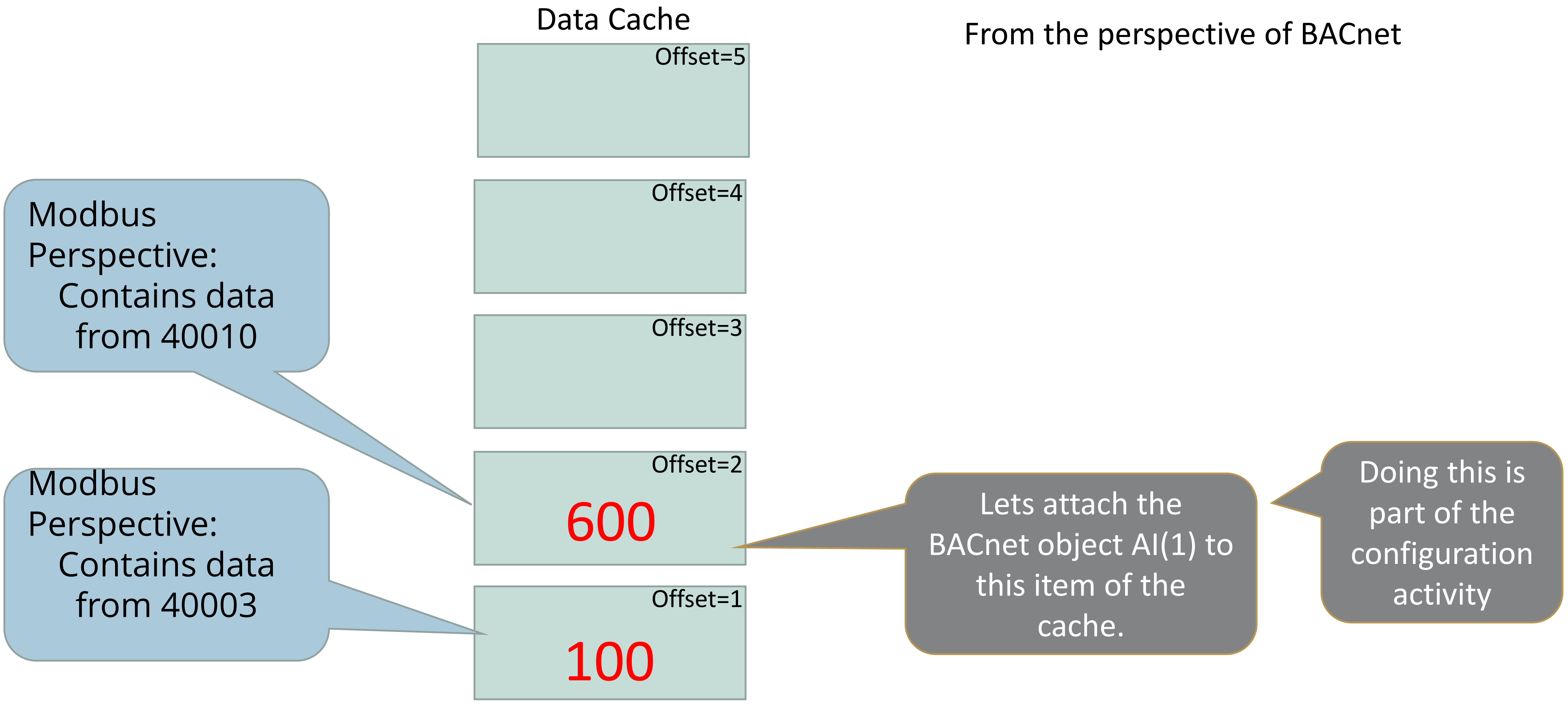
Modbus Perspective:
Contains data from 40010

Modbus Perspective:
Contains data from 40003

Data Type = Float
Number of Items = 5

WE need to create BACnet data objects

From the perspective of BACnet



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Modbus Perspective: Contains data from 40010

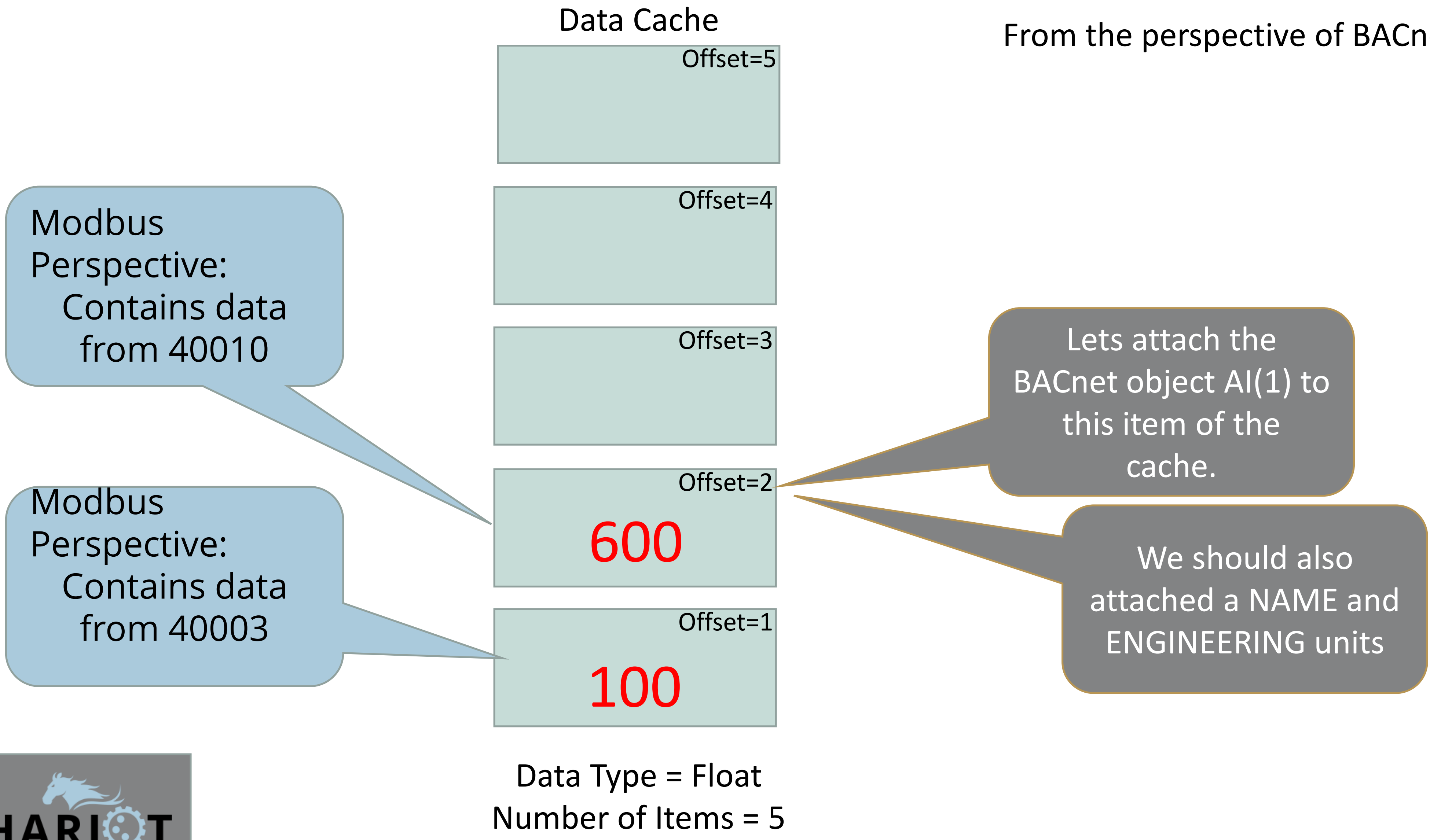
Modbus Perspective: Contains data from 40003

Lets attach the BACnet object AI(1) to this item of the cache.

Doing this is part of the configuration activity

Data Type = Float
Number of Items = 5

From the perspective of BACnet



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Data Type = Float
Number of Items = 5

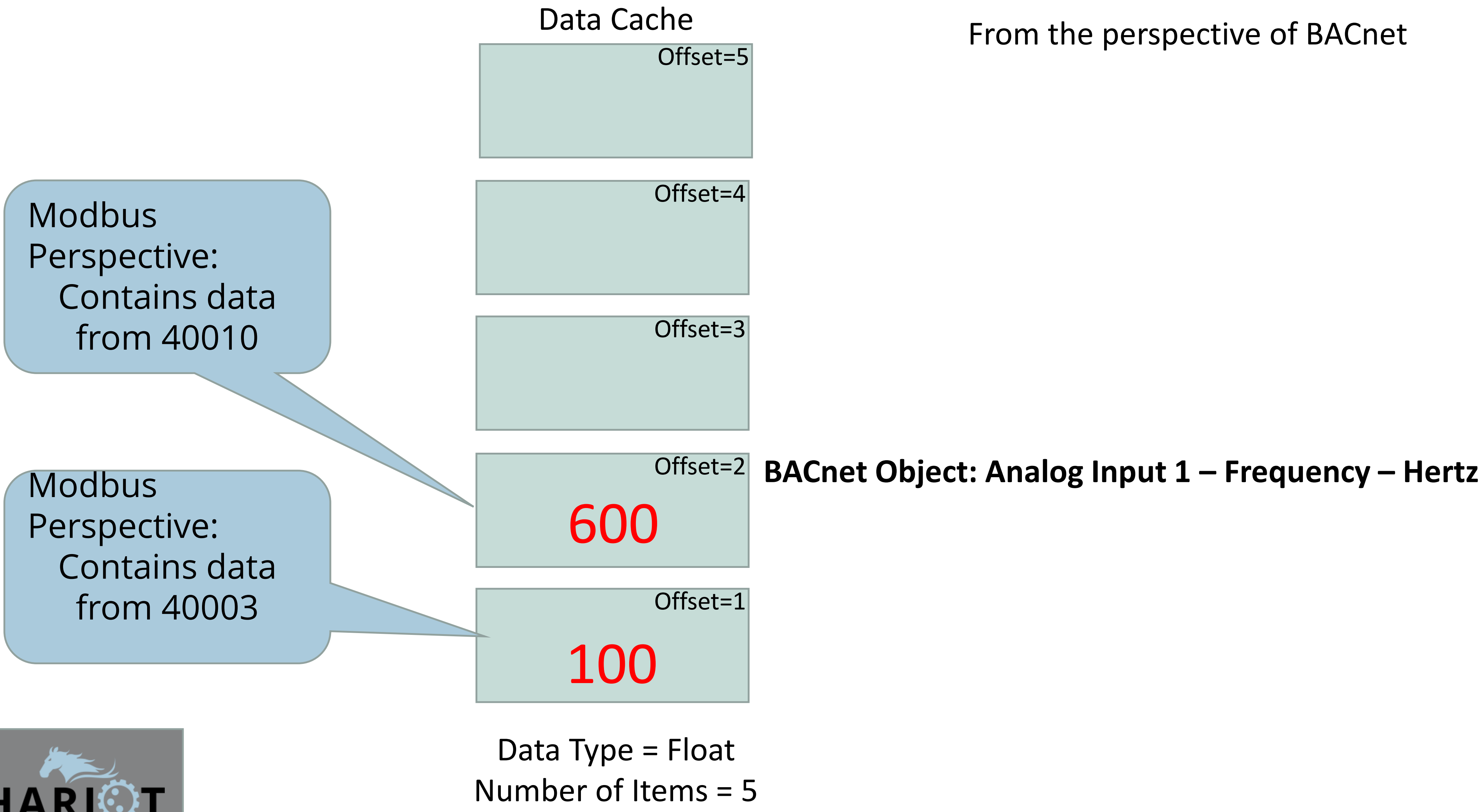
Modbus Perspective:
Contains data from 40010

Modbus Perspective:
Contains data from 40003

Lets attach the BACnet object AI(1) to this item of the cache.

We should also attached a NAME and ENGINEERING units

From the perspective of BACnet



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

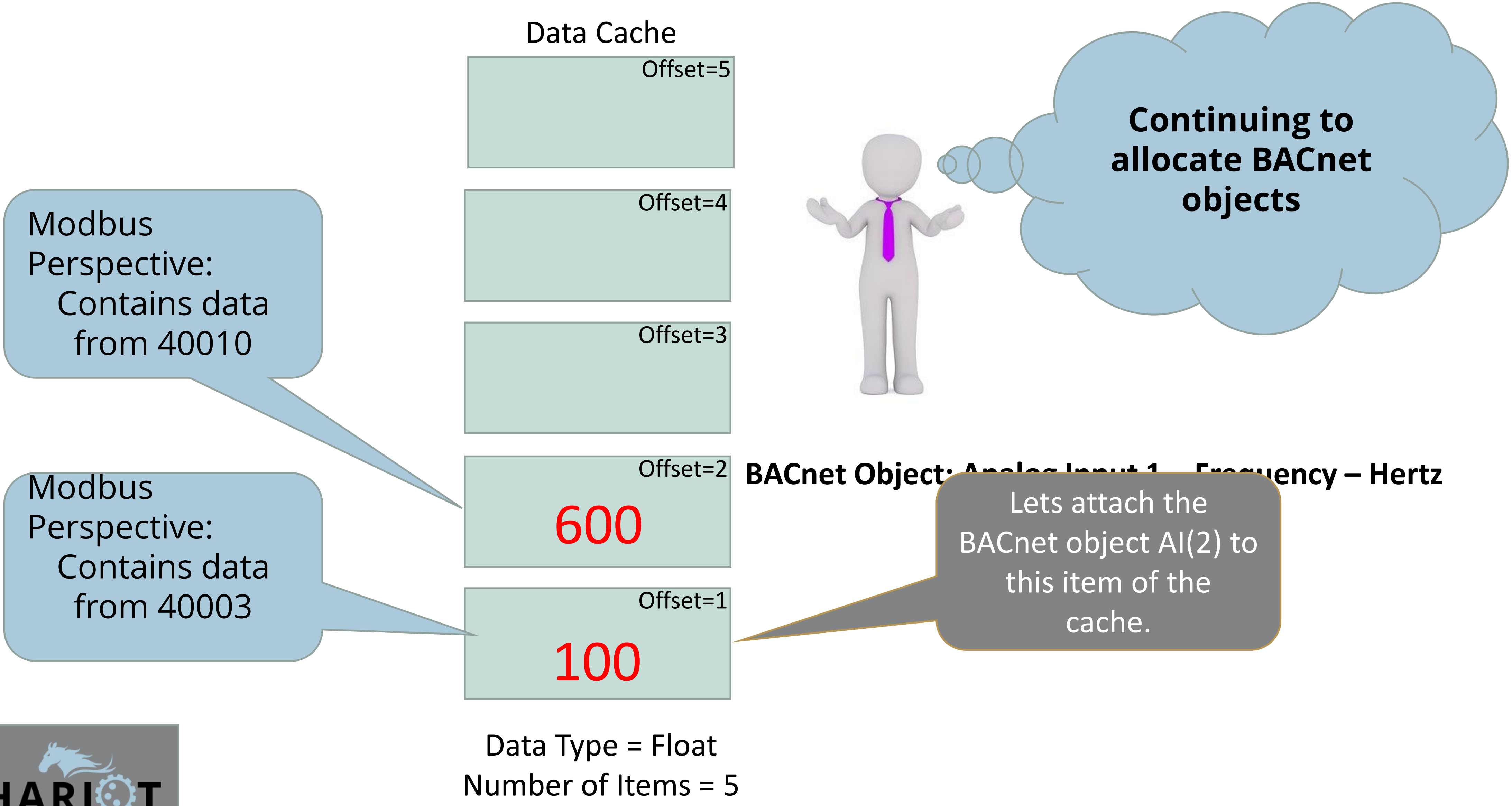
100

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

BACnet Object: Analog Input 1 – Frequency – Hertz

Data Type = Float
Number of Items = 5



Continuing to allocate BACnet objects

Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

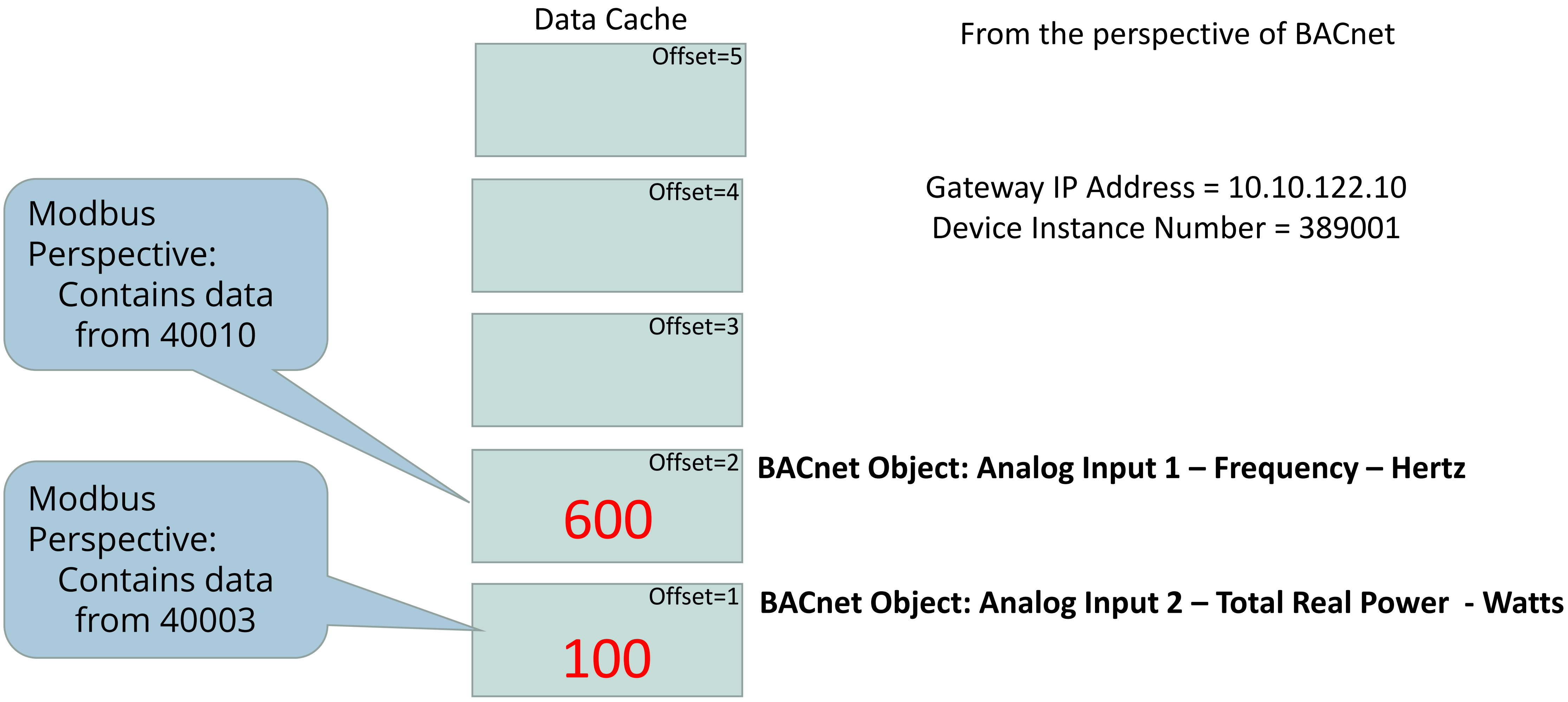
Data Type = Float
Number of Items = 5

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

BACnet Object: Analog Input 1 – Frequency – Hertz

Lets attach the BACnet object AI(2) to this item of the cache.



Data Cache

Offset=5

Offset=4

Offset=3

Offset=2

600

Offset=1

100

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

From the perspective of BACnet

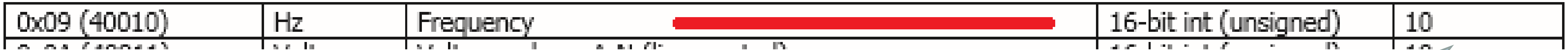
Gateway IP Address = 10.10.122.10
Device Instance Number = 389001

BACnet Object: Analog Input 1 – Frequency – Hertz

BACnet Object: Analog Input 2 – Total Real Power - Watts

Data Type = Float
Number of Items = 5

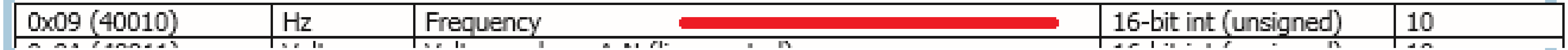
CHARIOT
BY CHIPKIN



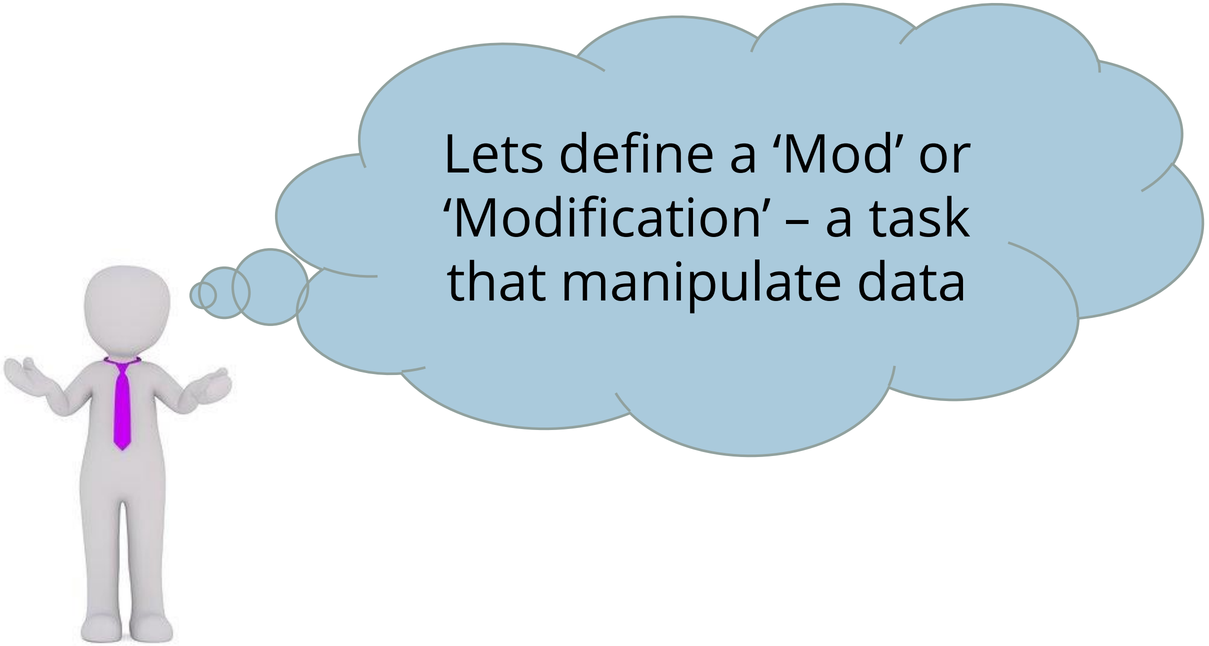
0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10

We never solved the scaling problem.

We see the scaling factor is 10.
Meaning if we read a value of 600 in register 40010 then we must divide by 10 to get the frequency of 60

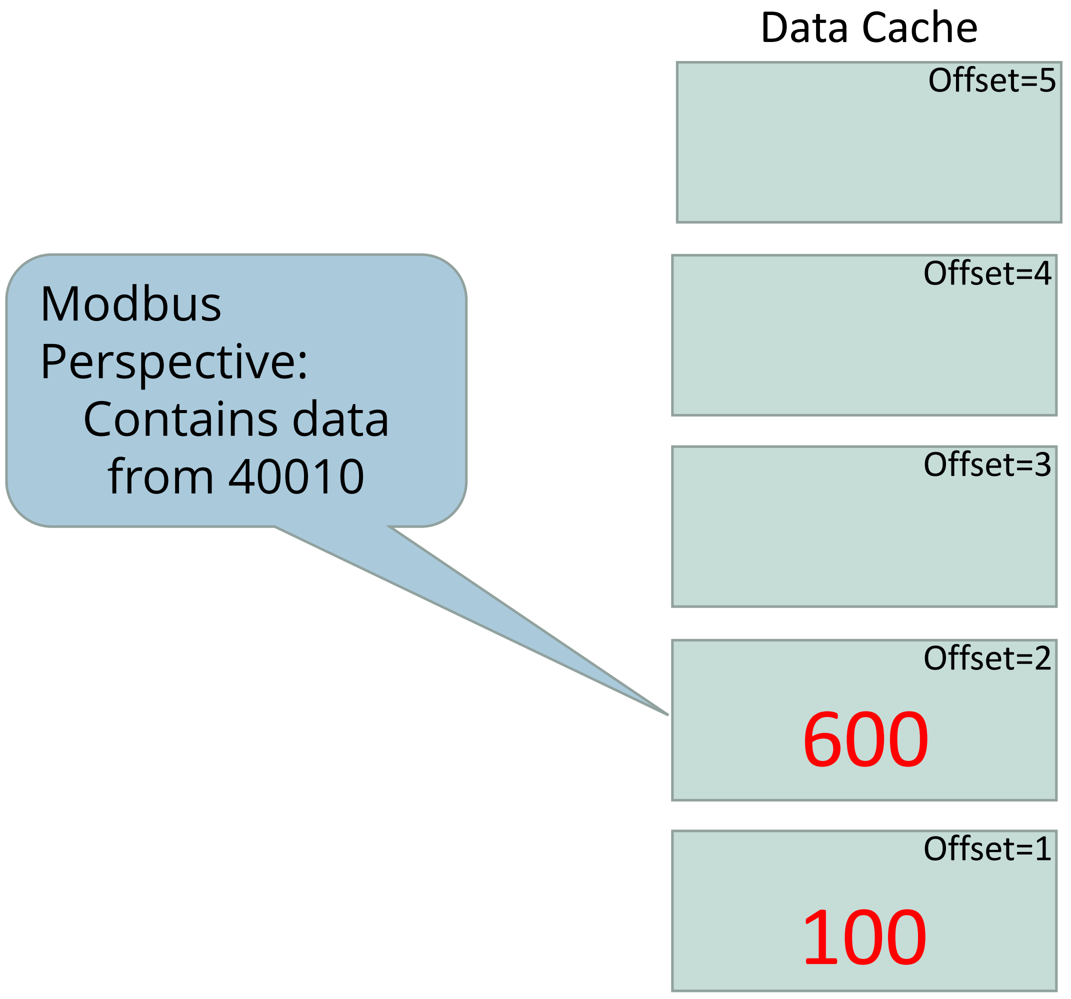


0x09 (40010)	Hz	Frequency	16-bit int (unsigned)	10



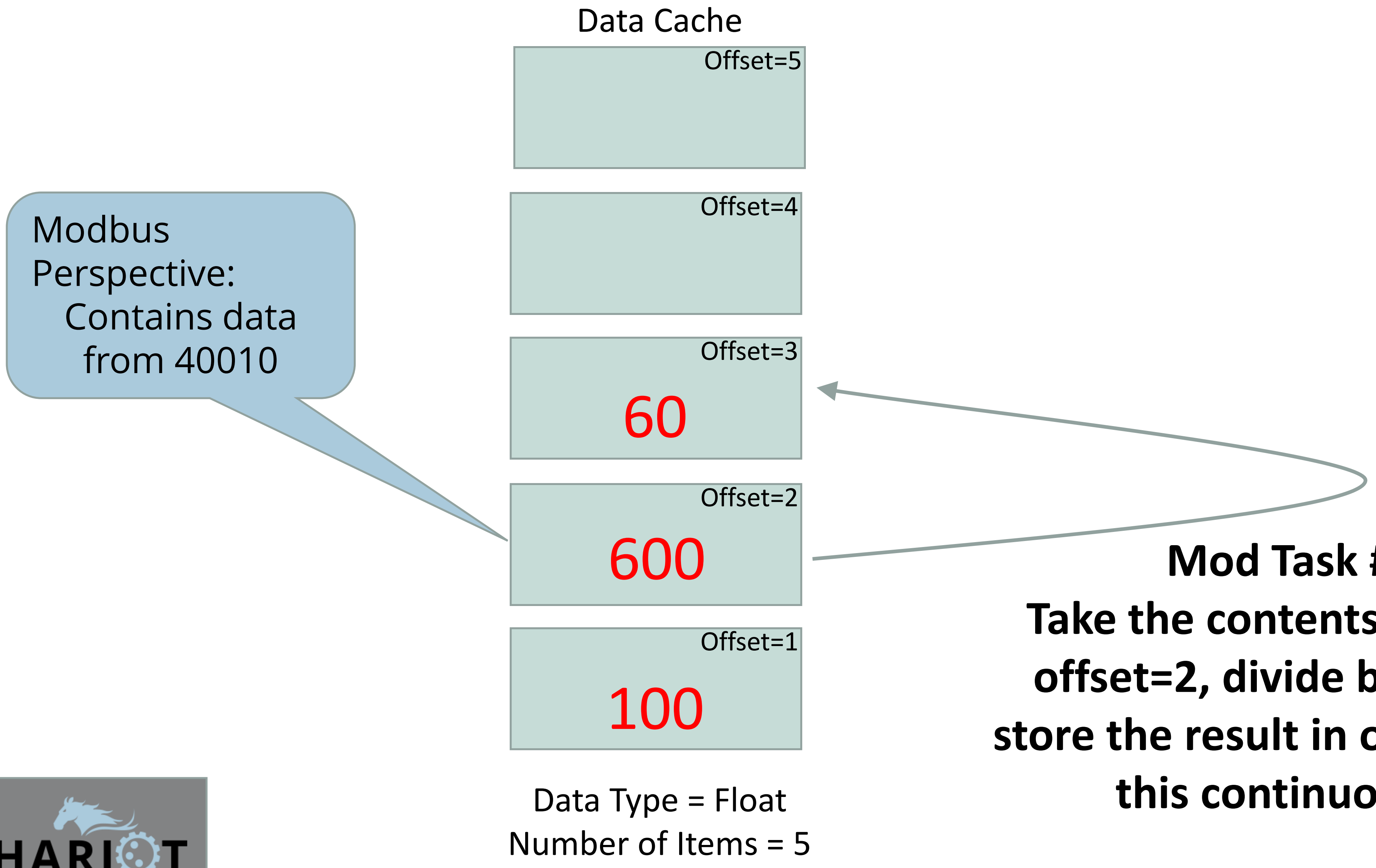
Lets define a ‘Mod’ or ‘Modification’ – a task that manipulate data

Mod Task #1
Take the contents of cache offset=2, divide by 10 and store the result in offset 3. Do this continuously.



BACnet Object: Analog Input 1 – Frequency – Hertz

Data Type = Float
Number of Items = 5



Data Cache

Offset=5

Offset=4

Offset=3

60

Offset=2

600

Offset=1

100

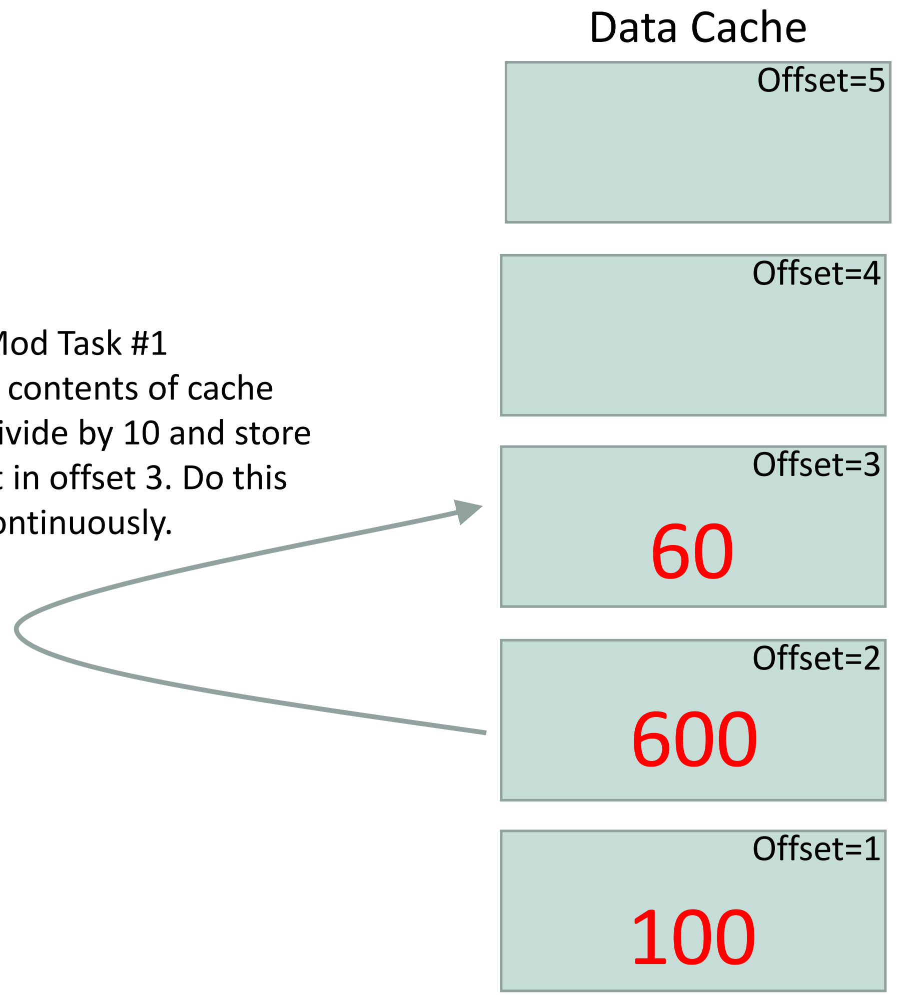
Data Type = Float
Number of Items = 5

Modbus Perspective:
Contains data from 40010

Mod Task #1

Take the contents of cache offset=2, divide by 10 and store the result in offset 3. Do this continuously.

Mod Task #1
Take the contents of cache offset=2, divide by 10 and store the result in offset 3. Do this continuously.



Data Cache

Offset=5

Offset=4

Offset=3
60

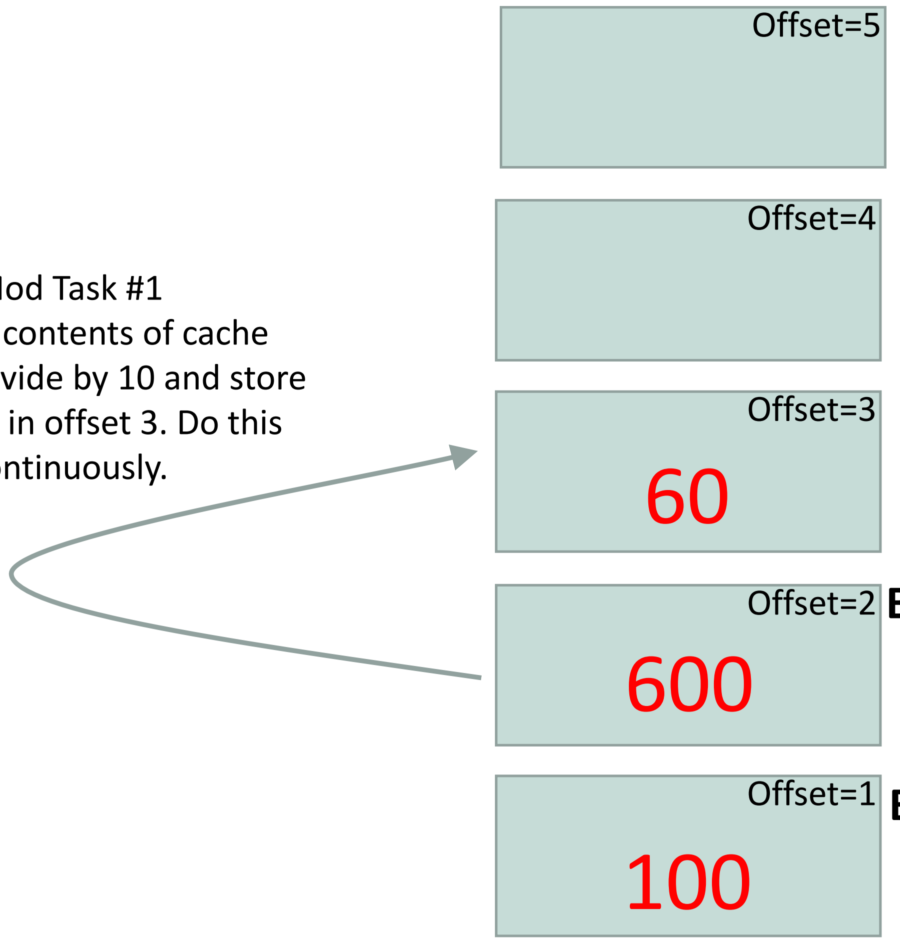
Offset=2
600

Offset=1
100

Data Type = Float
Number of Items = 5

CHARIOT
BY CHIPKIN

Data Cache



Offset=5

Offset=4

Offset=3

60

Offset=2

600

BACnet Object: Analog Input 1 – Frequency - Hertz

Offset=1

100

BACnet Object: Analog Input 2 – Total Real Power - Watts

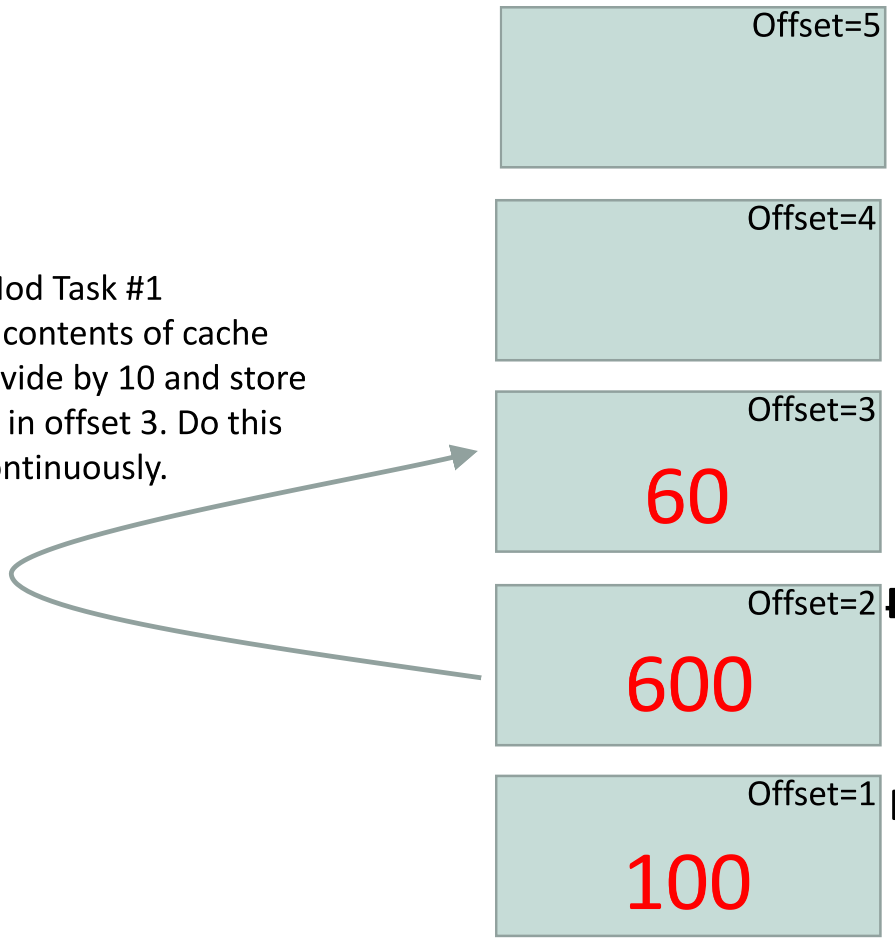
Mod Task #1
Take the contents of cache offset=2, divide by 10 and store the result in offset 3. Do this continuously.

It makes sense to change the BACnet mapping

Data Type = Float
Number of Items = 5

CHARIOT BY CHIPKIN

Data Cache



Offset=5

Offset=4

Offset=3

60

Offset=2 ~~BACnet Object: Analog Input 1 – Frequency - Hertz~~

600

Offset=1 **BACnet Object: Analog Input 2 – Total Real Power - Watts**

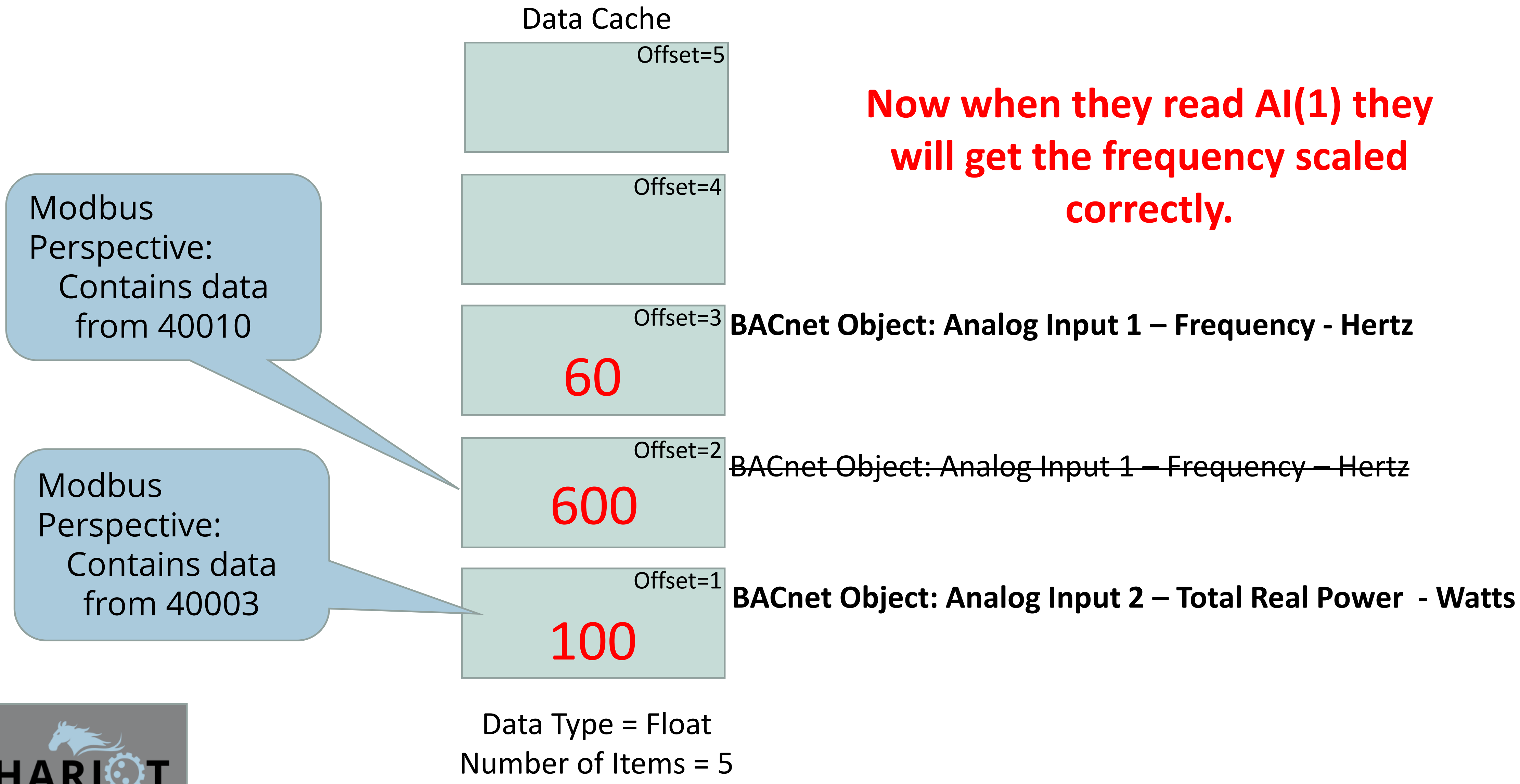
100

Mod Task #1
Take the contents of cache offset=2, divide by 10 and store the result in offset 3. Do this continuously.

It makes sense to change the BACnet mapping

Data Type = Float
Number of Items = 5

Now when they read AI(1) they will get the frequency scaled correctly.



Data Cache

Offset=5

Offset=4

Offset=3

60

BACnet Object: Analog Input 1 – Frequency - Hertz

Offset=2

600

~~BACnet Object: Analog Input 1 – Frequency – Hertz~~

Offset=1

100

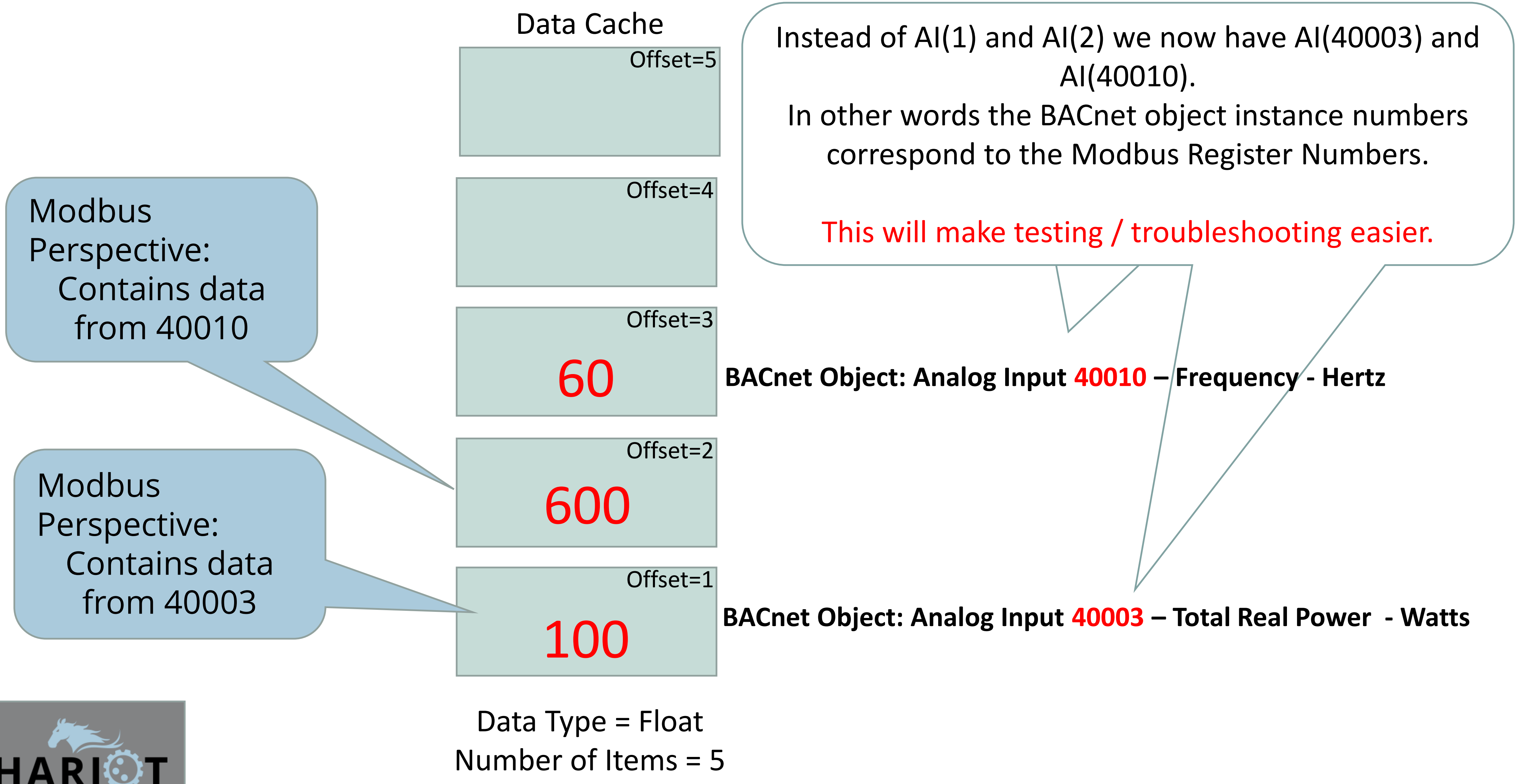
BACnet Object: Analog Input 2 – Total Real Power - Watts

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

Data Type = Float
Number of Items = 5

Lets make testing easier



Data Cache

Offset=5

Offset=4

Offset=3

60

Offset=2

600

Offset=1

100

Data Type = Float
Number of Items = 5

Modbus Perspective: Contains data from 40010

Modbus Perspective: Contains data from 40003

BACnet Object: Analog Input 40010 – Frequency - Hertz

BACnet Object: Analog Input 40003 – Total Real Power - Watts

Instead of AI(1) and AI(2) we now have AI(40003) and AI(40010).
In other words the BACnet object instance numbers correspond to the Modbus Register Numbers.

This will make testing / troubleshooting easier.

The End

2018 June

Contact Chipkin – chariot@chipkin.com