Science & Technology Facilities Council

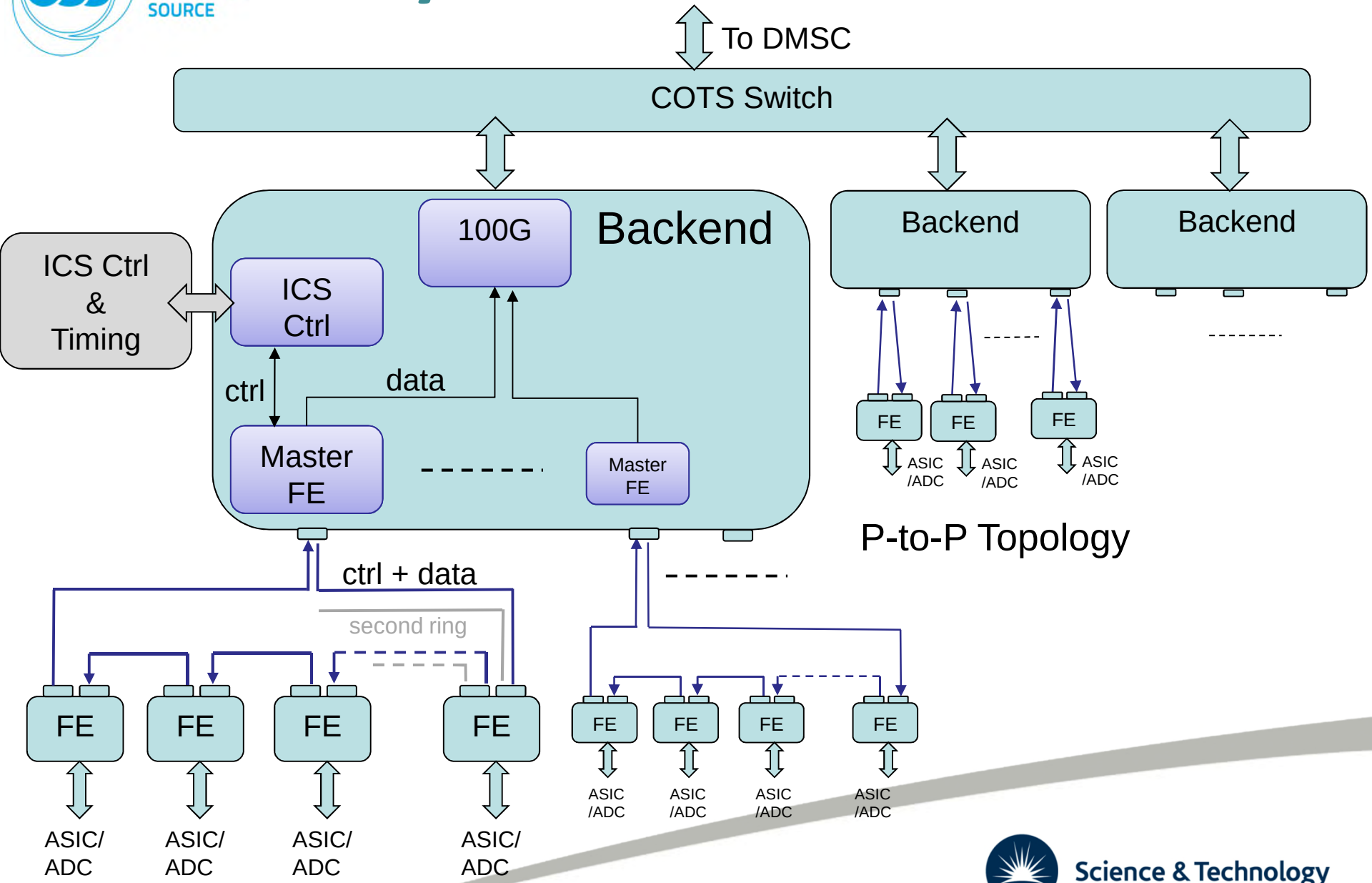
Slow Control and Bulk Data Meeting 21,22 Jan'19

Dr. Harry Walton

Rutherford Appleton Laboratory, UK

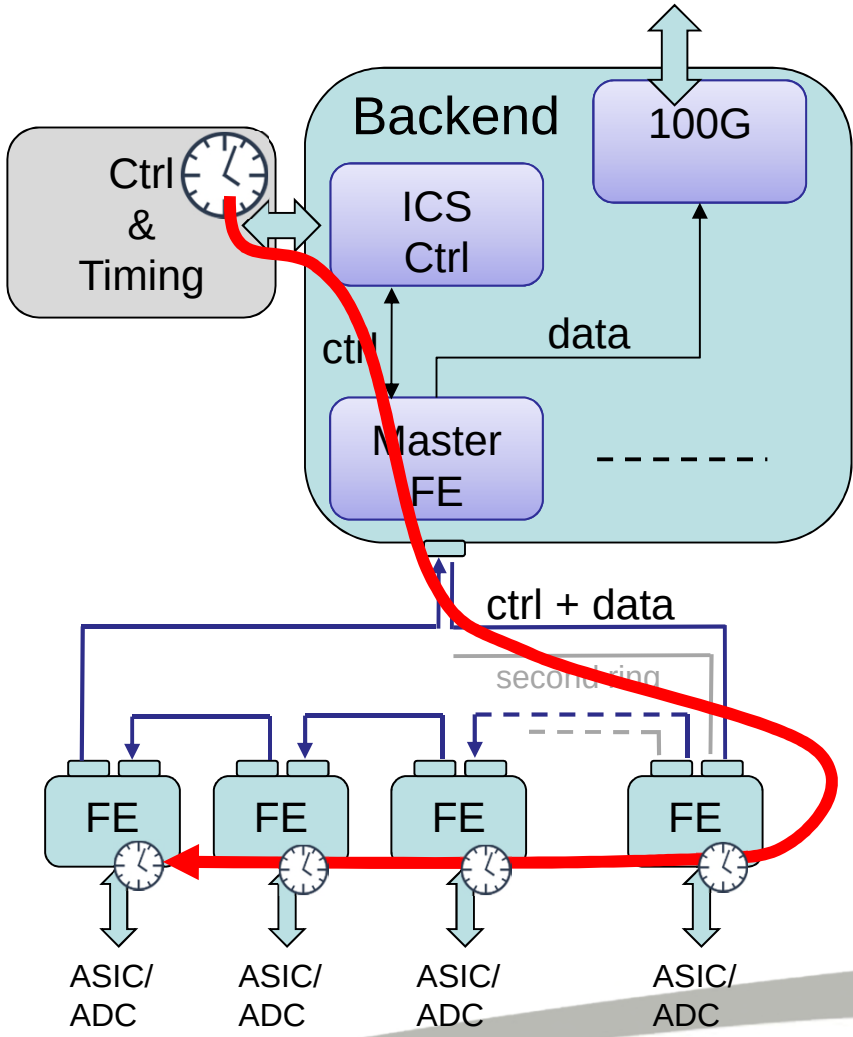
EUROPEAN SPALLATION SOURCE

System Architecture



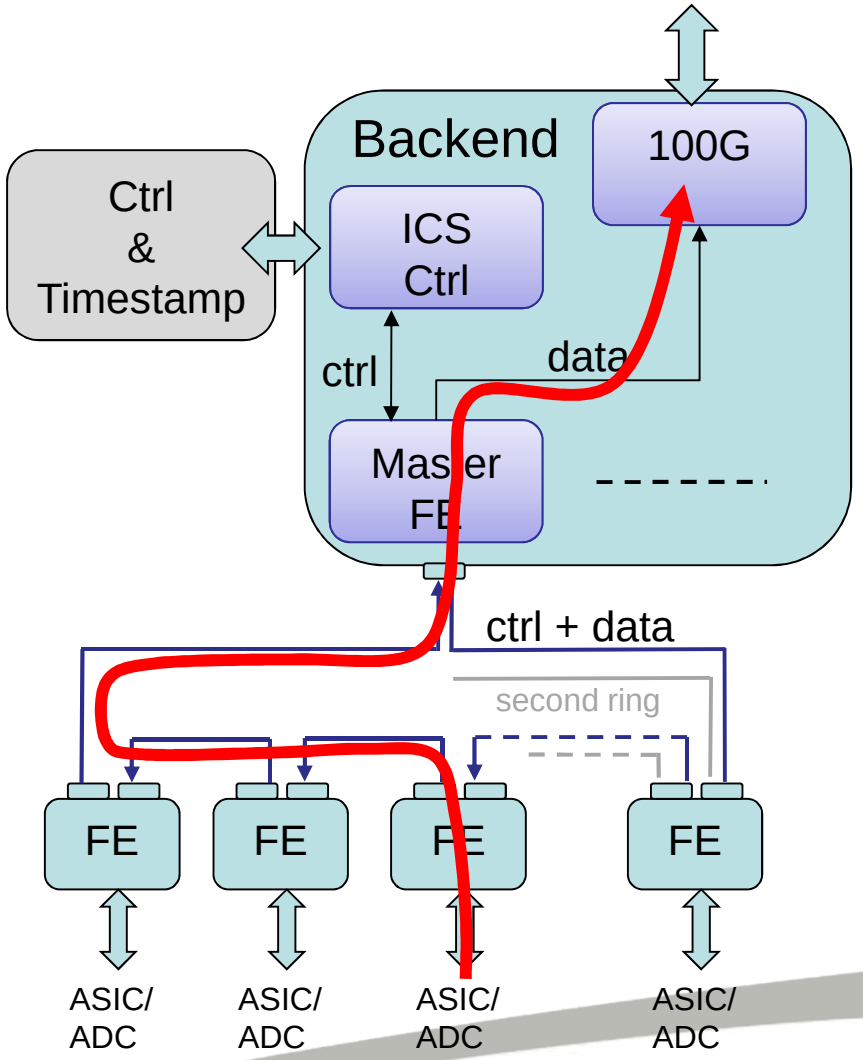
Front End Functions

- Acquire accurate timestamp.
- Collect digitized (timestamped) Bulk data, and downstream it to the BE.
- Receive & Return Memory Mapped Slow Control data (e.g. ADC-register W/R's)



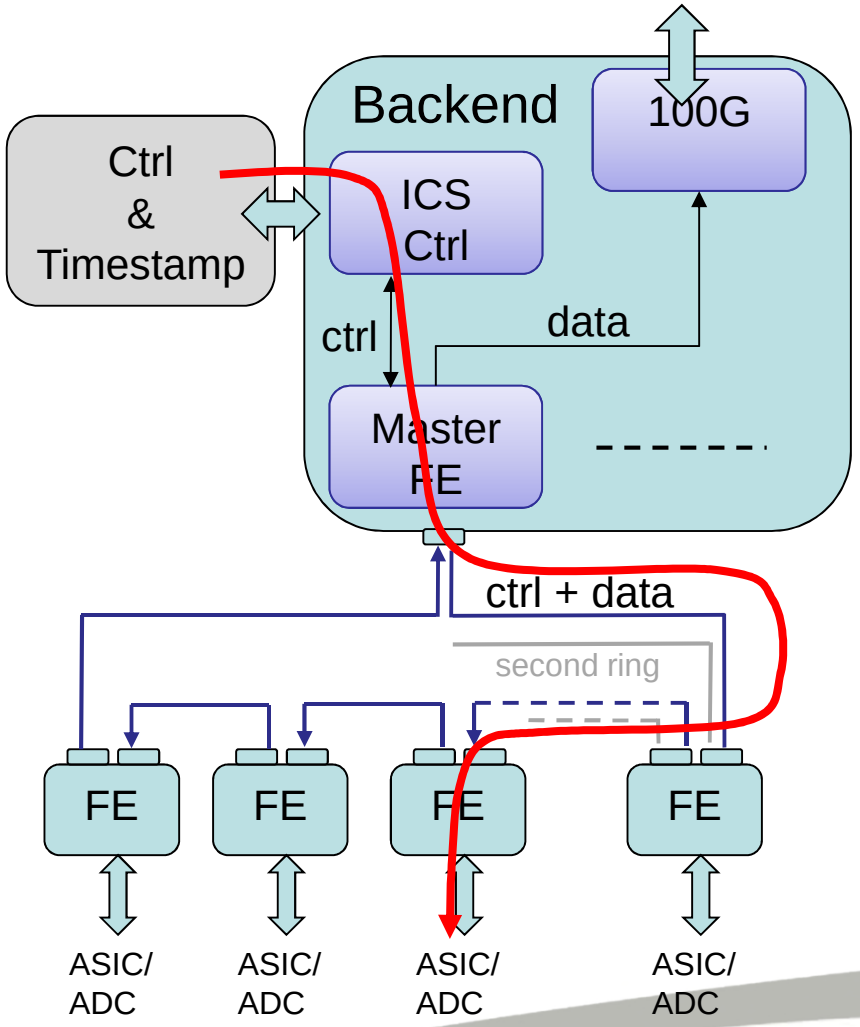
Front End Functions

- Acquire accurate timestamp.
- Collect digitized (timestamped) Bulk , and downstream it to the BE.
- Receive & Return Slow Control data (e.g. ADC-register W/R's)

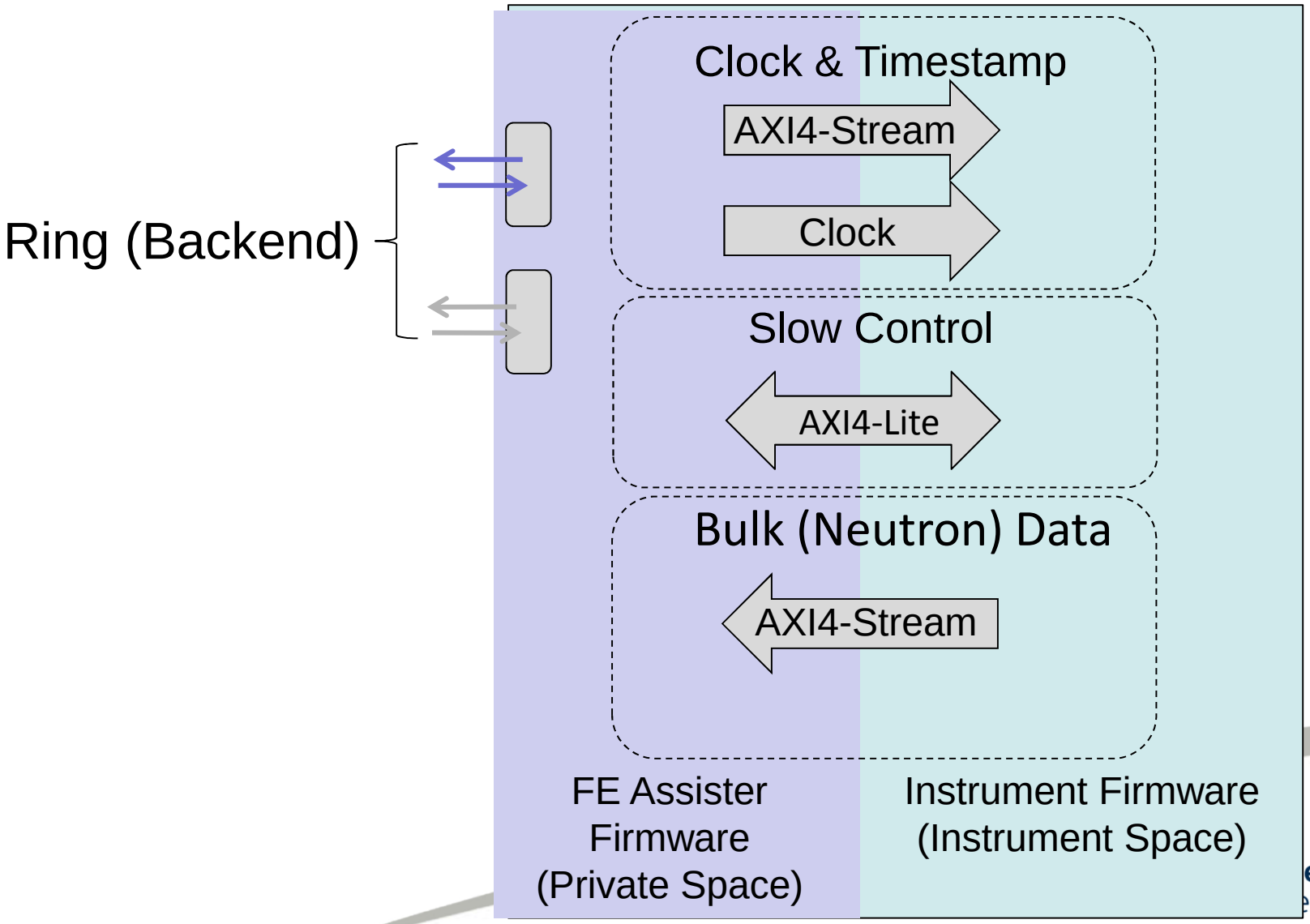


Front End (FE) Functions

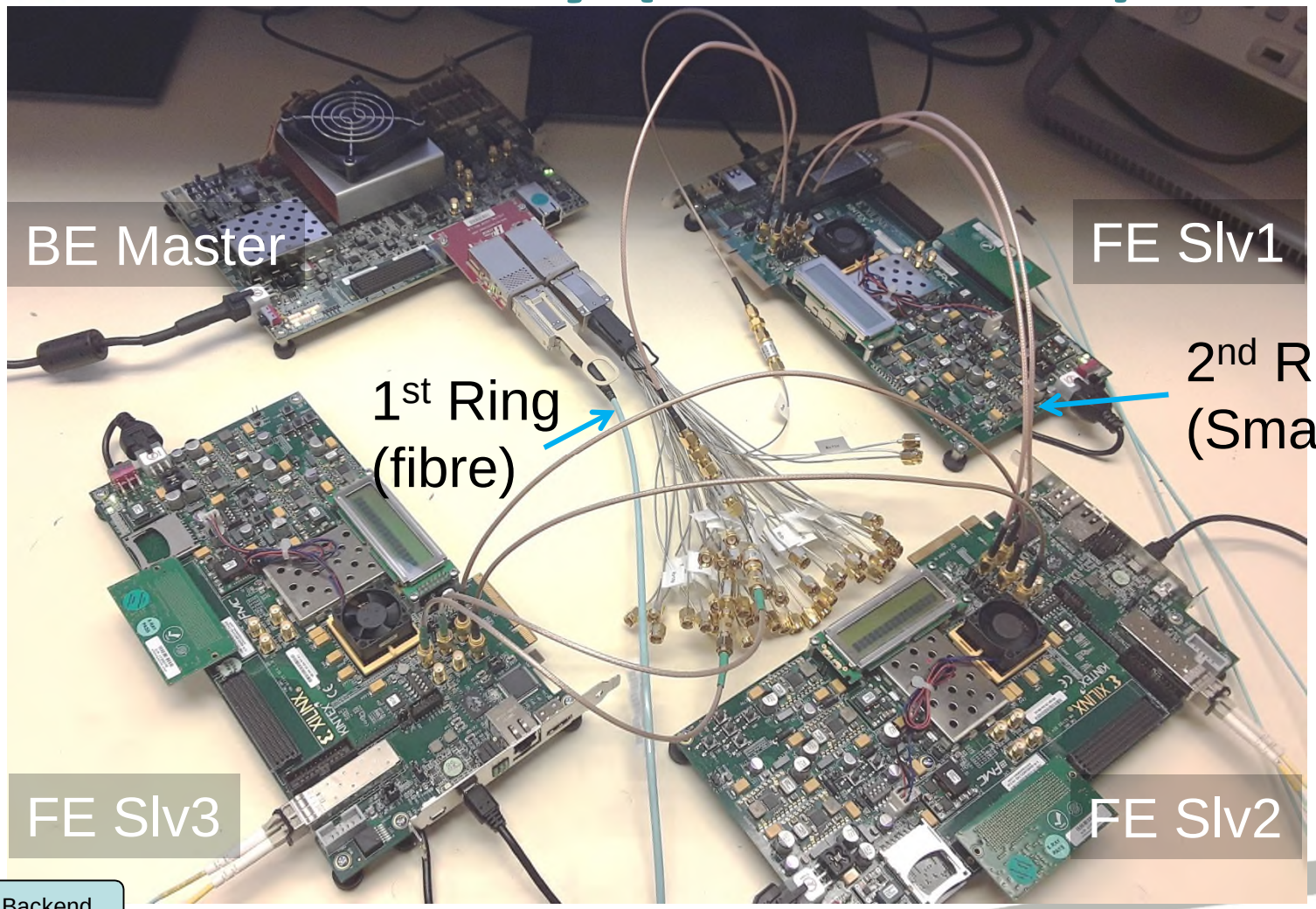
- Acquire accurate timestamp.
- Collect digitized (timestamped) Bulk data, and downstream it to the BE.
- Receive & Return Memory Mapped Slow Control data (e.g. ADC-register W/R's)

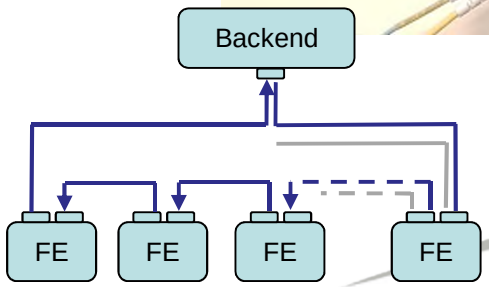


Front End User Interface



Previously (IKON 2018)





Science & Technology Facilities Council

EUROPEAN SPALLATION SOURCE

Previously shown proof-of-concept

- Acquire a timestamp. ✔
- Collect digitized (timestamped) Bulk data, and downstream it to the BE. ✔
- Receive & Return Memory Mapped Slow Control data (e.g. ADC-register W/R's) ✔

Science & Technology Facilities Council

Previous Limitations

- V. limited Slow Control & Diagnostics during ring bring-up
- Dual-ring & P-to-P operation

Today's update

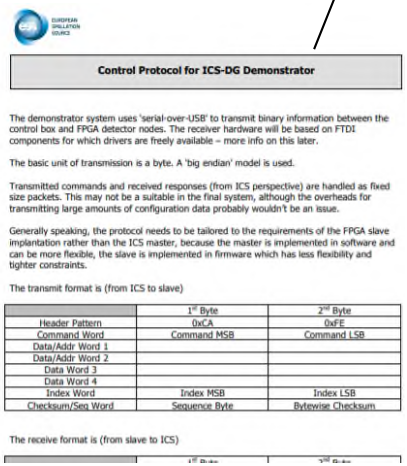
- Integration & test of larger (multi-FE) ring
- Integration with ESS 88MHz Timing

Awaiting requisite H/W

“Getting to Science”

Ctrl I/f at present:

Python



Control Protocol for ICS-DG Demonstrator

The demonstrator system uses 'serial-over-USB' to transmit binary information between the control box and FPGA detector nodes. The receiver hardware will be based on FTDI components for which drivers are freely available – more info on this later.

The basic unit of transmission is a byte. A 'big endian' model is used.

Transmitted commands and received responses (from ICS perspective) are handled as fixed size packets. This may not be a suitable in the final system, although the overheads for transmitting large amounts of configuration data probably wouldn't be an issue.

Generally speaking, the protocol needs to be tailored to the requirements of the FPGA slave implantation rather than the ICS master, because the master is implemented in software and can be more flexible, the slave is implemented in firmware which has less flexibility and tighter constraints.

The transmit format is (from ICS to slave)

	1st Byte	2nd Byte
Header Pattern	0xCA	0xFE
Command Word	Command MSB	Command LSB
Data/Addr Word 1		
Data/Addr Word 2		
Data Word 3		
Data Word 4		
Index Word	Index MSB	Index LSB
Checksum/Seq Word	Sequence Byte	Bytewise Checksum

The receive format is (from slave to ICS)

Python over UART in accordance with ESS doc. *draft_protocol.pdf* ‘Control Protocol for ICS DG Demonstrator’

Power-on

Bring-up Ring

Config. Detectors

Run

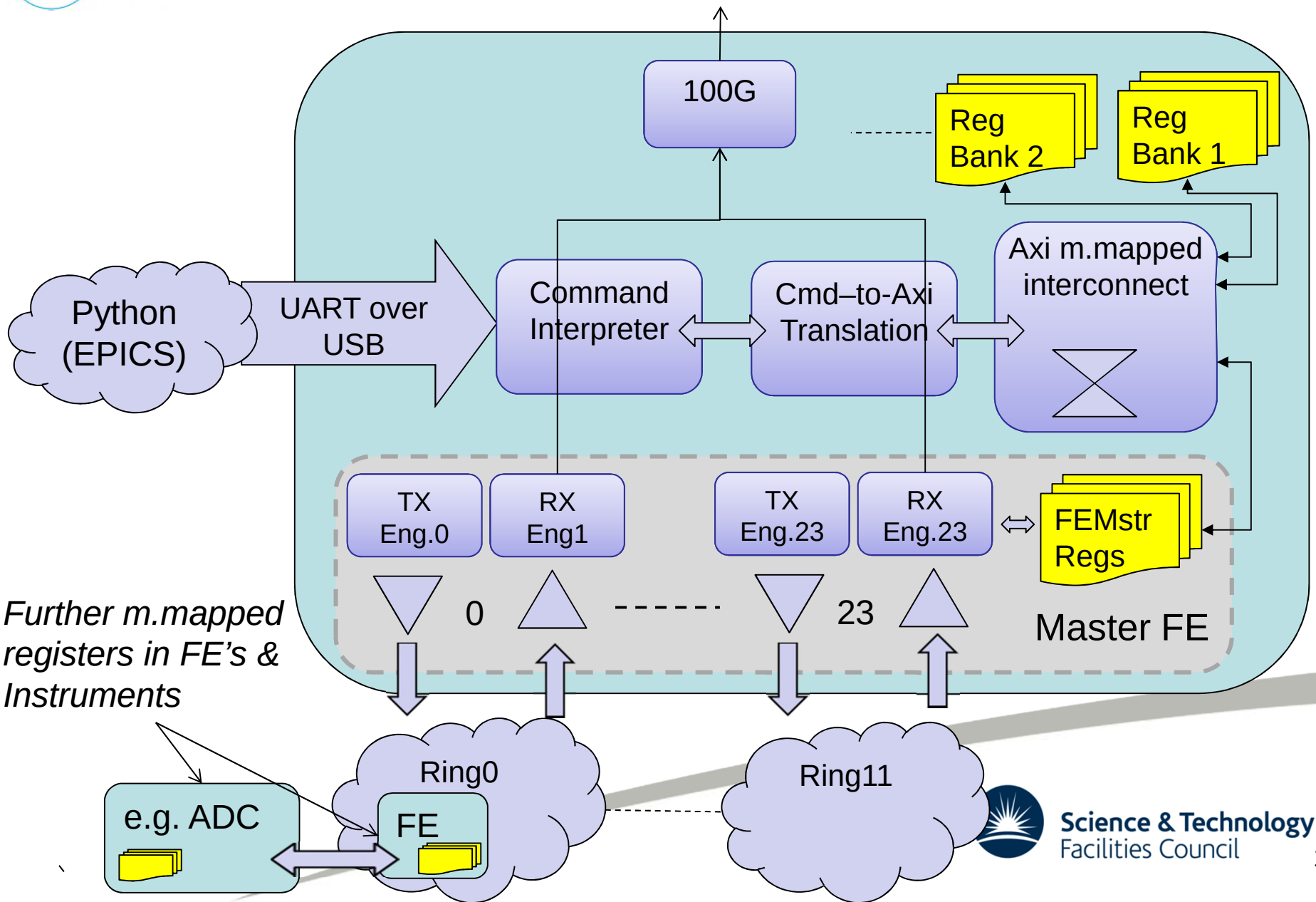
Science & Technology Facilities Council

Memory Mapped System

ESS Demonstrator Controls Address Space (for 24 Pt-2-Pt)

Address Range	Size	Region	Comments
0000'0000 – 3FFF'FFFF	1 GB	BE	
4000'0000 – 47FF'FFFF	128 MB	Ring 0	Local + User spaces
4800'0000 – 4FFF'FFFF	128 MB	Ring 1	
		...	
		...	
F800'0000 – FFFF'FFFF	128 MB	Ring 23	Max 24 Pt-2-Pt

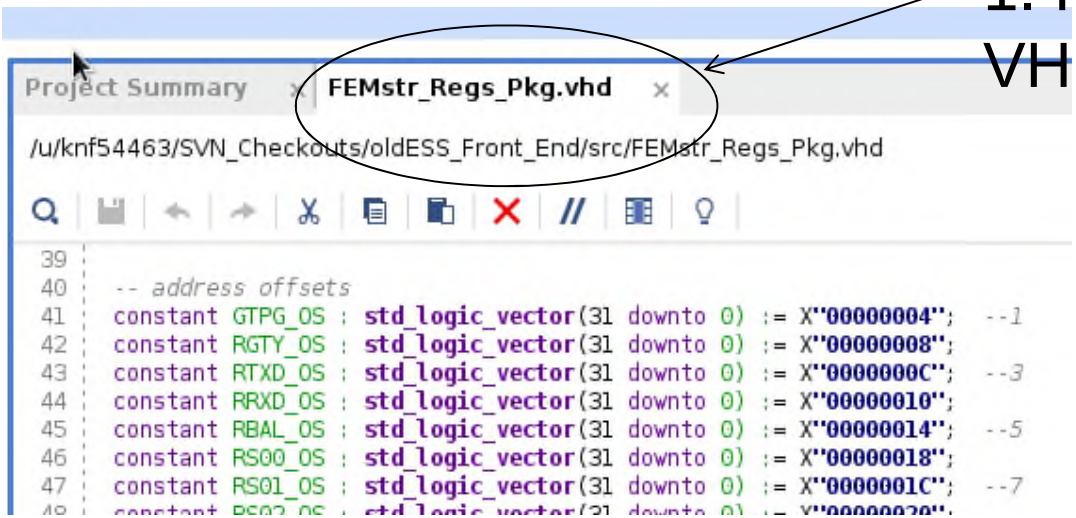
New Version Master Front End



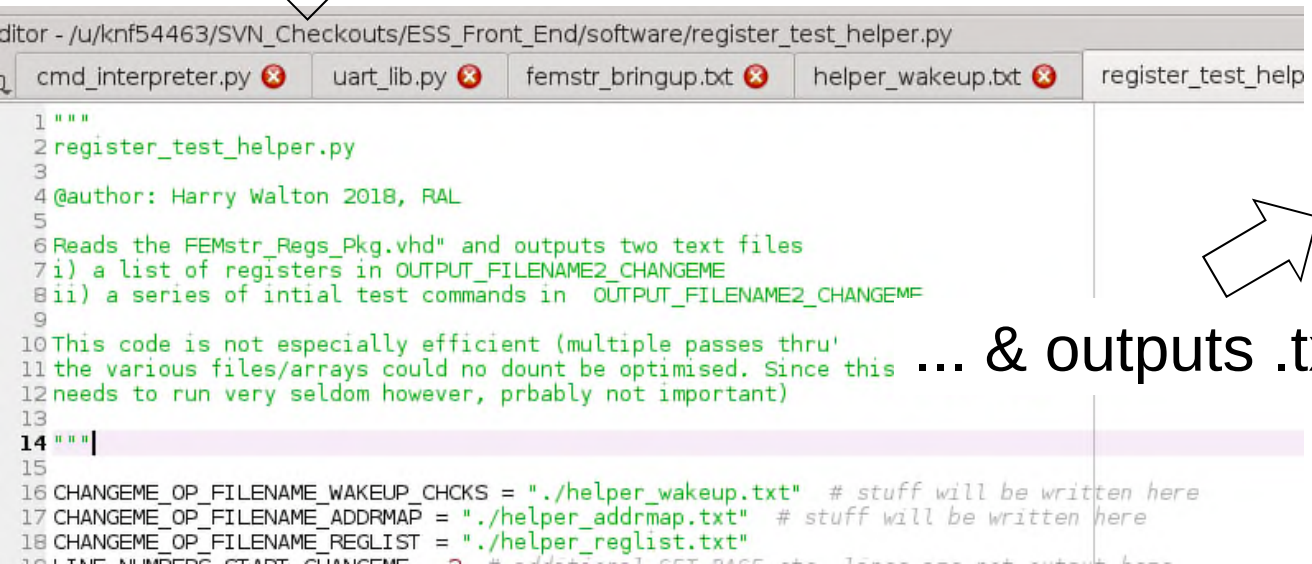
Science & Technology Facilities Council

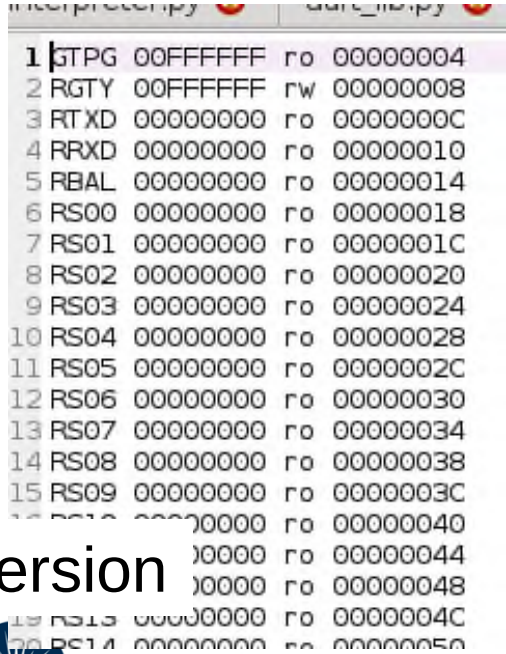
Register Definition

1. Registers defined in original VHDL



2. Python helper-program parses...



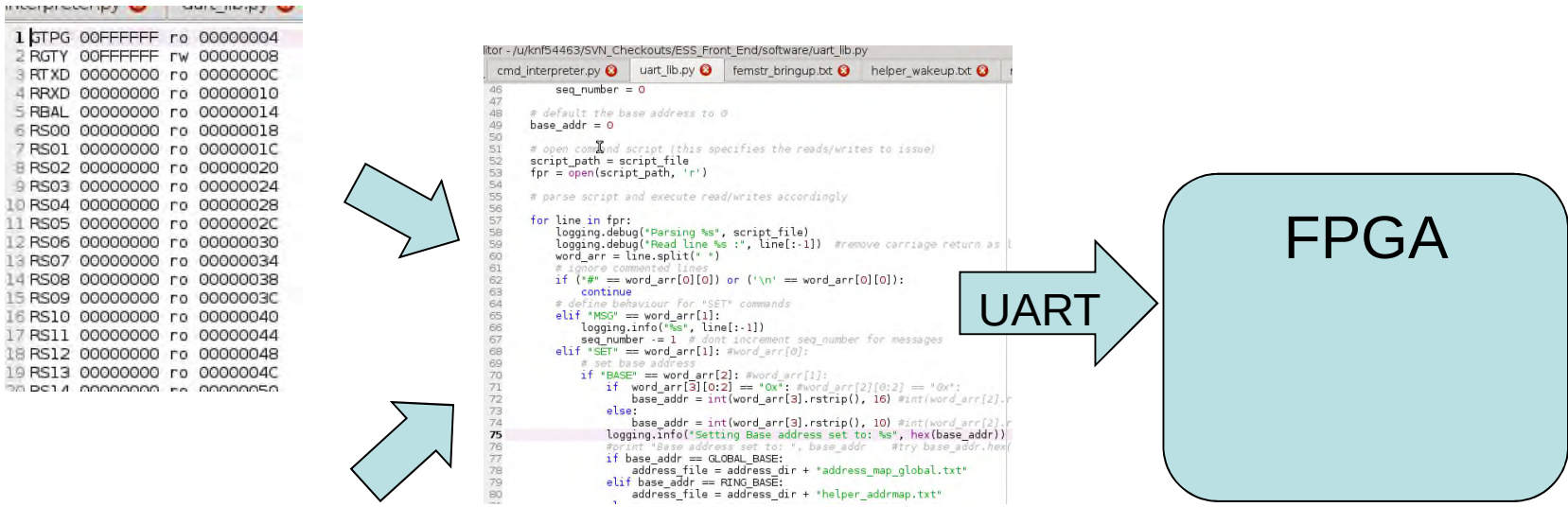


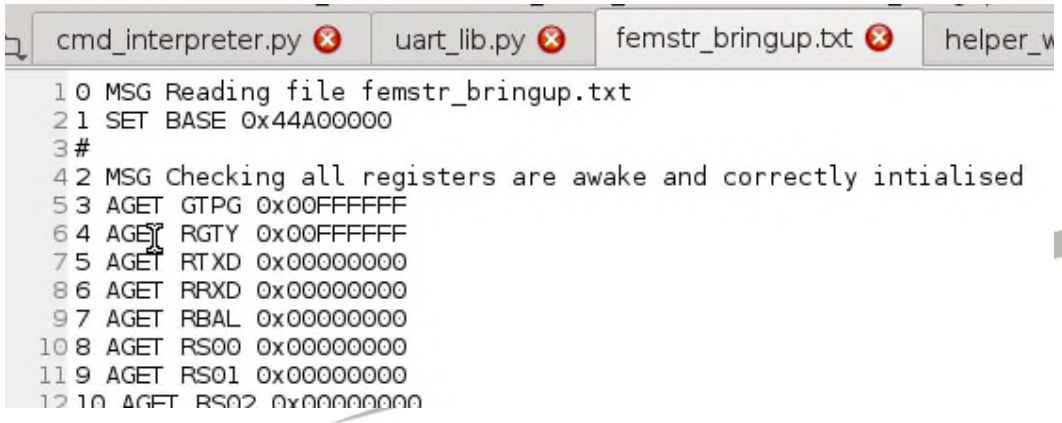
... & outputs .txt version

Science & Technology Facilities Council

Python Bring-up

3. Python takes *register.txt* + a *bring-up-script,* and drives the UART





Science & Technology Facilities Council

Bring-Up Script Snippet

(Example : powering up 3 of the available 24 “rings”)

print message to python console

```
129 #
130 121 MSG Reset all GTY
131 122 SET RGTY 0xFFFFFFFF
132 123 AGET RGTY 0xFFFFFFFF
133 124 MSG Un-reset 1st, 4th and 7th GTYs  (0xB6 = 1011_0110)
134 125 SET RGTY 0xFFFFFFB6
135 126 AGET RGTY 0xFFFFFFB6
136 #
137 127 MSG Check reset cntrlr sees 1st, 4th & 7th GTY out of rst (0x49=0100_1001)
```

poke 0xFF...B6 to RGTY reg.

check (peek) RGTY reg. now holds 0xFF...B6

(Future nice-to-haves:
- More commands: Read-Set-a-bit-Writeback etc.
- Named bitmasks etc.)

Running the bring-up script

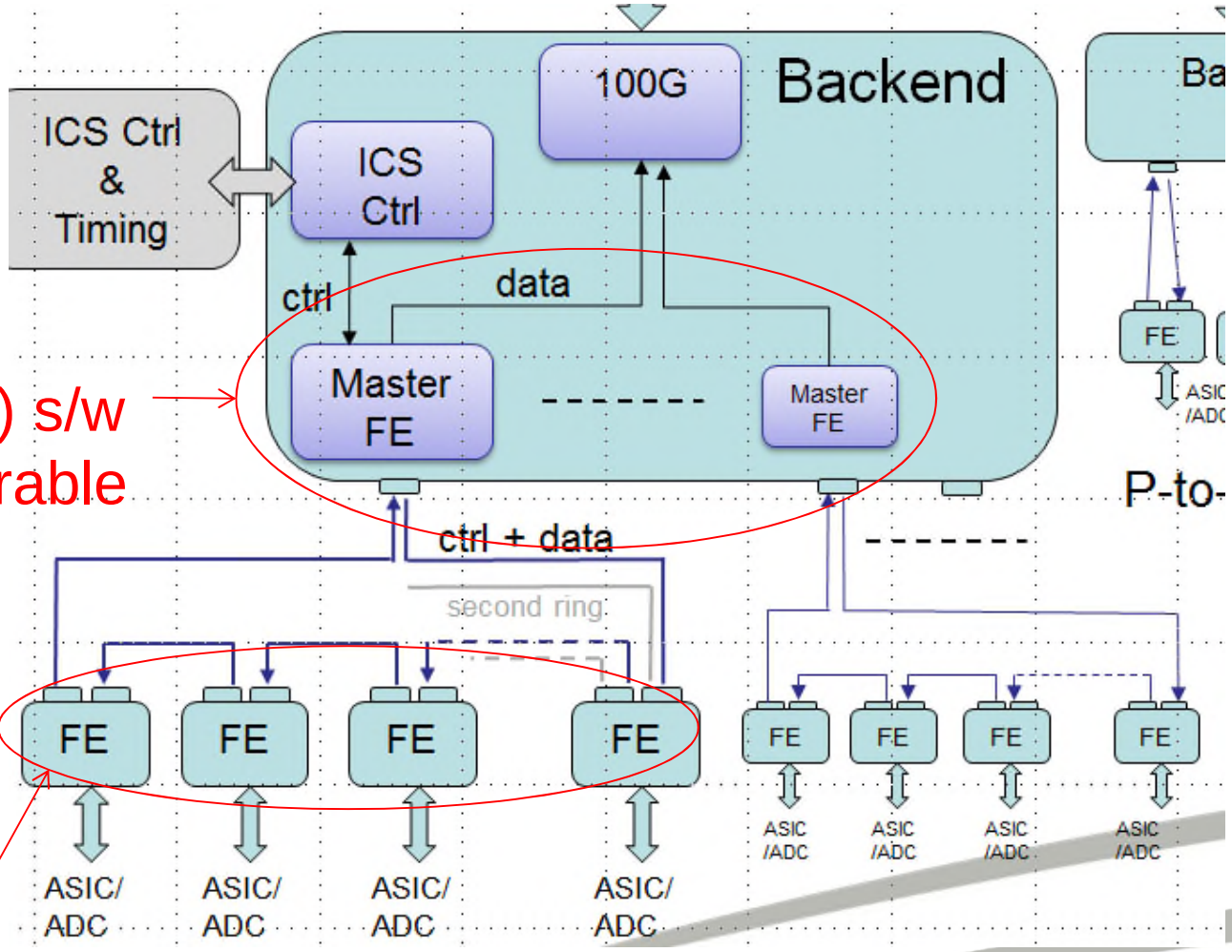
```
118 MSG ...So un-powerdown respective CPl
119 SET PDPL 0xFFFFFFB6
120 AGET PDPL 0xFFFFFFB6
#
121 MSG Reset all GTY
122 SET RGTY 0xFFFFFFFF
123 AGET RGTY 0xFFFFFFFF
124 MSG Un-reset 1st, 4th and 7th GTYs
125 SET RGTY 0xFFFFFFB6
126 AGET RGTY 0xFFFFFFB6
#
127 MSG Check reset cntrllr sees 1st, 4tl
128 AGET RTXD 0x00000049
129 AGET RRXD 0x00000049
#
130 MSG Check the (state-machine) status
131 MSG ...Each nibble of TCSx encodes s
132 MSG ...include TXIDLEst=0000, TXBGUP
132 AGET TCS0 0x00000000
132 AGET TCS1 0x00000000
```

```
INFO: 118 MSG connects to cable-T-hell-9, ar
INFO: 118 MSG ...So un-powerdown respective
INFO: o.k.
INFO: 121 MSG Reset all GTY
INFO: o.k.
INFO: 124 MSG Un-reset 1st, 4th and 7th GTYs
INFO: o.k.
INFO: 127 MSG Check reset cntrllr sees 1st,
INFO: o.k.
INFO: o.k.
INFO: 130 MSG Check the (state-machine) stat
INFO: 131 MSG ...Each nibble of TCSx encodes
INFO: 132 MSG ...include TXIDLEst=0000, TXBC
INFO: o.k.
INFO: o.k.
INFO: o.k.
INFO: 158 MSG Put all TX engines in TXIDLEs
INFO: 133 MSG In the TXIDLEst the TX engine
INFO: 134 MSG ...interrogating all the TSxx
INFO: o.k.
```

```
2019-01-15 13:40:03,148 uart_lib.py, 182 DEBUG ******** Sequence No.= 119
2019-01-15 13:40:03,149 uart_lib.py, 58 DEBUG Parsing femstr_bringup.txt
2019-01-15 13:40:03,149 uart_lib.py, 59 DEBUG Read line # :
2019-01-15 13:40:03,149 uart_lib.py, 58 DEBUG Parsing femstr_bringup.txt
2019-01-15 13:40:03,149 uart_lib.py, 59 DEBUG Read line 121 MSG Reset all GTY :
2019-01-15 13:40:03,149 uart_lib.py, 66 INFO 121 MSG Reset all GTY
2019-01-15 13:40:03,150 uart_lib.py, 182 DEBUG ******** Sequence No.= 119
2019-01-15 13:40:03,150 uart_lib.py, 58 DEBUG Parsing femstr_bringup.txt
2019-01-15 13:40:03,150 uart_lib.py, 59 DEBUG Read line 122 SET RGTY 0xFFFFFFFF :
2019-01-15 13:40:03,150 uart_lib.py, 124 DEBUG Attempting write to RGTY = 0x44a00008 with 0xffffffff
2019-01-15 13:40:03,151 uart_lib.py, 231 DEBUG Setting write start address to 0x44a00008
2019-01-15 13:40:03,153 uart_lib.py, 238 DEBUG Writing data 0xffffffff
2019-01-15 13:40:03,155 uart_lib.py, 182 DEBUG ******** Sequence No.= 120
2019-01-15 13:40:03,155 uart_lib.py, 58 DEBUG Parsing femstr_bringup.txt
2019-01-15 13:40:03,156 uart_lib.py, 59 DEBUG Read line 123 AGET RGTY 0xFFFFFFFF :
2019-01-15 13:40:03,156 uart_lib.py, 168 DEBUG Attempting read-with-assert from RGTY = 0x44a00008
2019-01-15 13:40:03,156 uart_lib.py, 255 DEBUG Setting read start address to 0x44a00008
2019-01-15 13:40:03,161 uart_lib.py, 170 DEBUG Read value 0xffffffff
```

Bring-up script input → GUI Console output + log.txt output

Status & Next Steps



Now, (largely) s/w configurable

Next, adapt FE RTL similarly for s/w bring-up

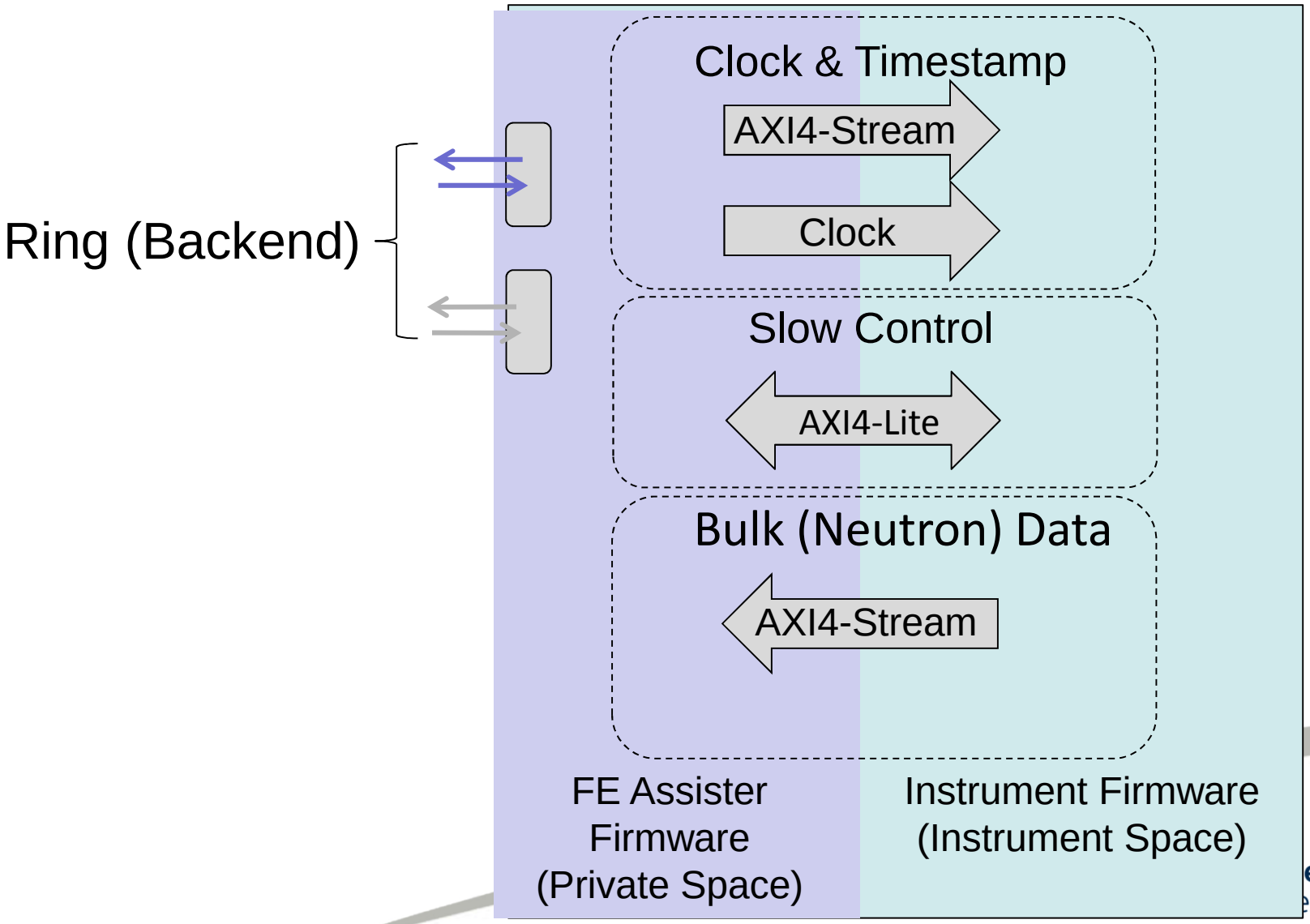
Science & Technology Facilities Council

EUROPEAN SPALLATION SOURCE

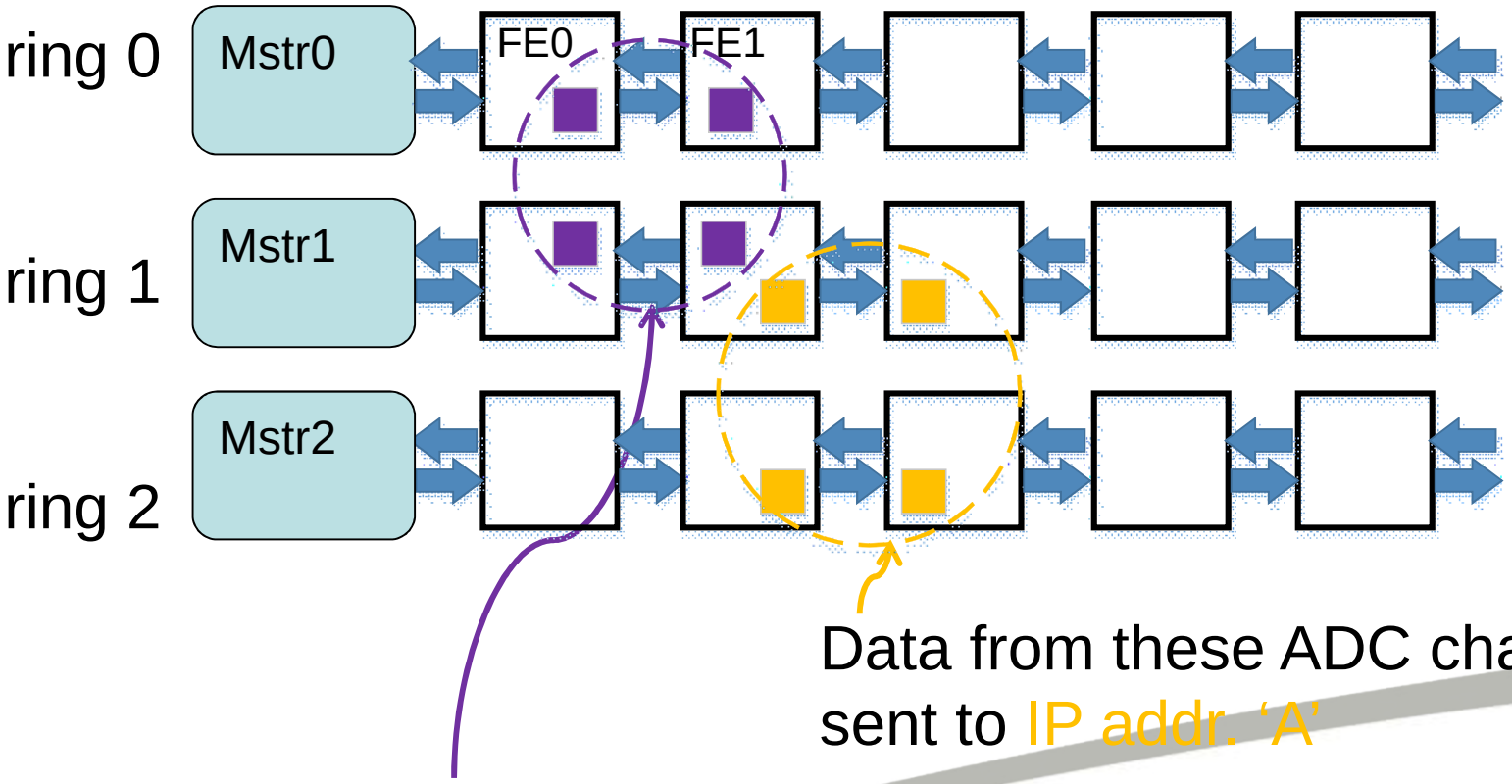
Bulk Data

Science & Technology Facilities Council

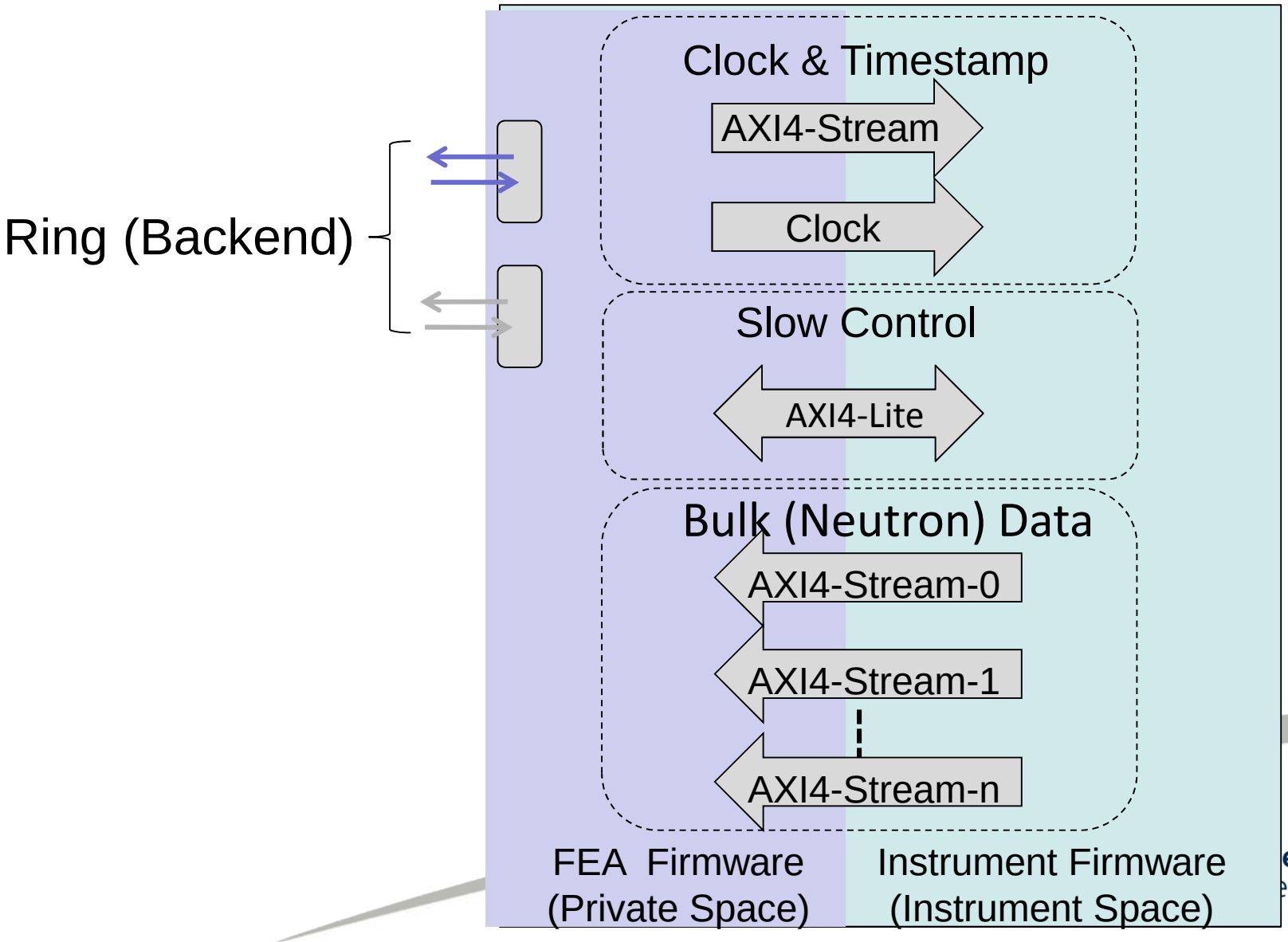
Front End User Interface



New understanding : Bulk Data Assigned to IP Addr. based on ADC channel



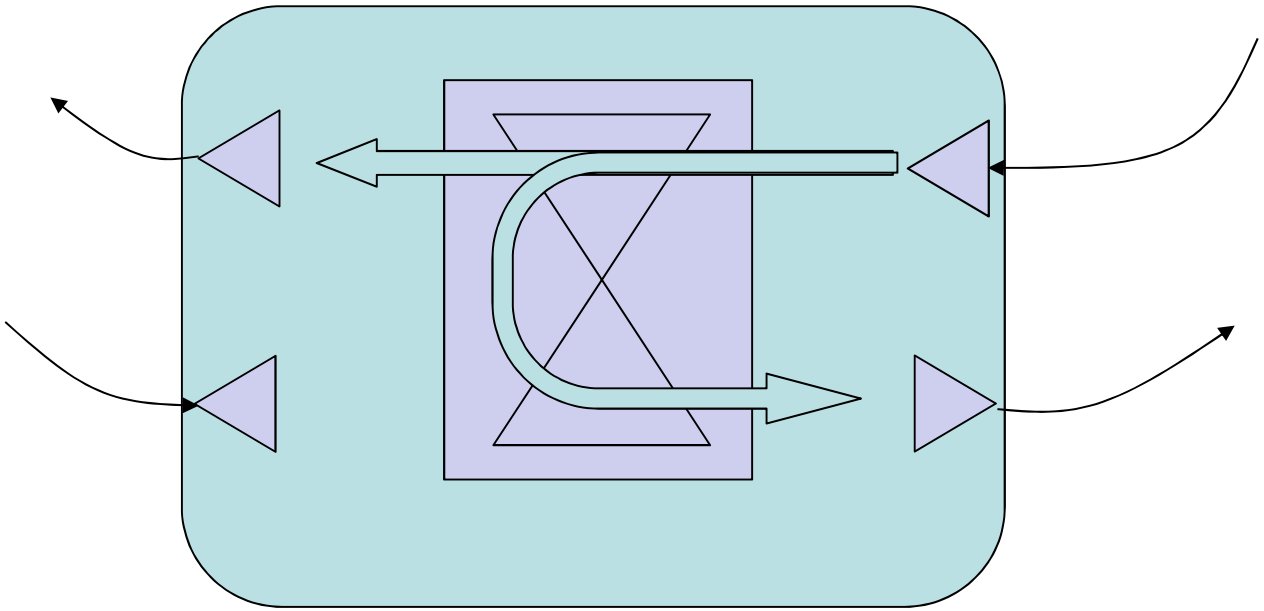
New Front End User Interface incorporating multiple data streams



EUROPEAN SPALLATION SOURCE

Ring Re-configuation

FE for crossbar



Science & Technology Facilities Council