“MAN-IN-THE-SCADA”:

Anatomy of Data Integrity Attacks in Industrial Control Systems

Marina Krotofil & Chris Sistrunk

black hat ASIA 2017
MARCH 28-31, 2017
MARINA BAY SANDS / SINGAPORE

About us

MK

Specialization:
Process Control

- Mostly on offence side
- 7 years in process control security research
- On and Off 13 years in security

CS

Specialization:
Power Sector

- Mostly on defense
- >10 years experience in power engineering
- 8 years in security

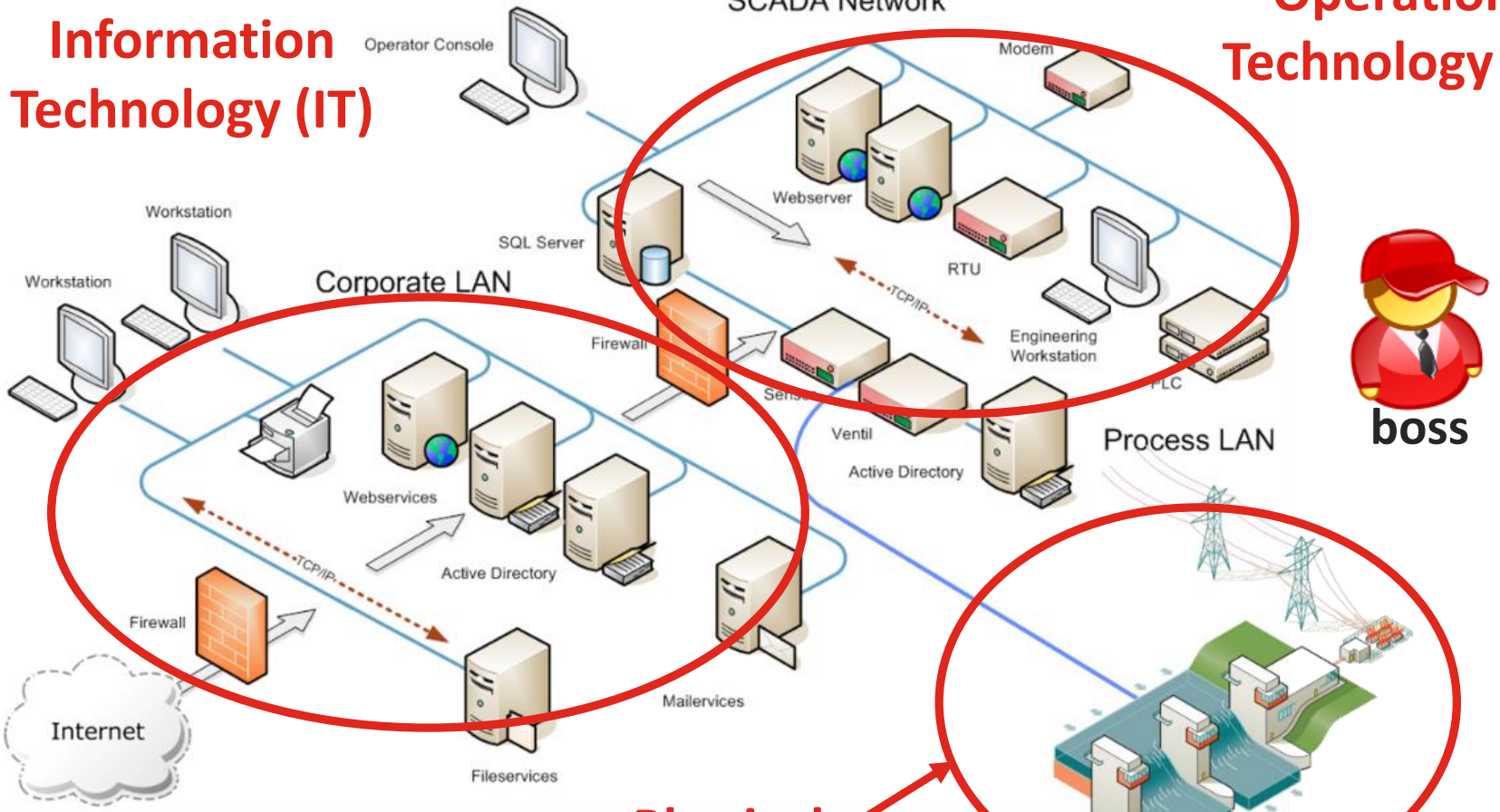
Setting Context

Industrial Control Systems and Cyber-Physical Hacking

Industrial Control Systems

Helpless desk

Information Technology (IT)



Operational Technology (OT)

Know how to run Win95

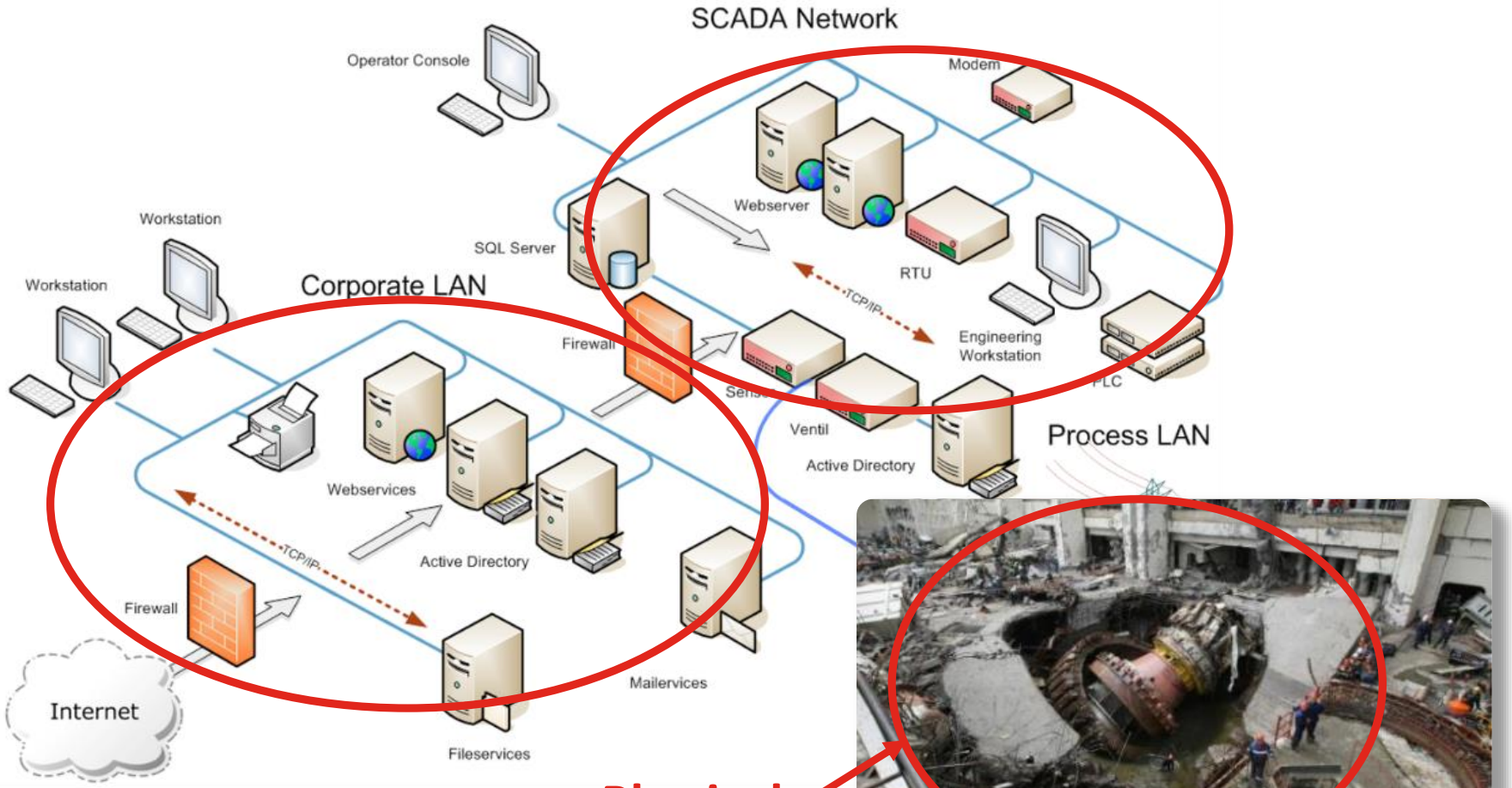
boss

boss

Physical application

Always blamed

Cyber-Physical Hacking



Challenging assumptions

Man-in-the-SCADA

Most frequently assumed scenario

OPERATOR

OPERATOR CONSOLE

CONTROL SYSTEM

PROCESS

Why?

- Insecurity by design of majority of industrial protocols
- Mechanics of MITM attack is well understood and tons of tools are readily available (almost Plug&Play)
- We simply DON’T KNOW BETTER (yet)

Let's look into the packet (2)

1	0.000000	192.168.101.14	192.168.101.…	EGD	75 Data Msg: ExchangeID=0x00000002, RequestID=00141
2	0.000000	192.168.101.14	192.168.101.…	EGD	94 Data Msg: ExchangeID=0x00000001, RequestID=00141
3	0.999977	192.168.101.14	192.168.101.…	EGD	75 Data Msg: ExchangeID=0x00000002, RequestID=00142
4	0.999977	192.168.101.14	192.168.101.…	EGD	94 Data Msg: ExchangeID=0x00000001, RequestID=00142
5	1.999952	192.168.101.14	192.168.101.…	EGD	94 Data Msg: ExchangeID=0x00000001, RequestID=00143
6	1.999952	192.168.101.14	192.168.101.…	EGD	75 Data Msg: ExchangeID=0x00000002, RequestID=00143
7	2.999917	192.168.101.14	192.168.101.…	EGD	94 Data Msg: ExchangeID=0x00000001, RequestID=00144

```
Version: 1
RequestID: 142
ProducerID: 192.168.101.14
ExchangeID: 0x00000001
Timestamp:
Status: No error, data consumed (2)
ConfigSignature: 257
Reserved: 1313518940
Data (20 bytes)
    Data: 040000005d60e04fddefdf4fd9387a4ea53c6c4f
    [Length: 20]
```

???

Ugh :-(
I need protocol parsers. I knew it!

```
0000  ff ff ff ff ff ff 2c 27  d7 19 d0 23 08 00 45 00   ......,' ...#..E.
0010  00 50 0b 3d 00 00 05 11  5e 02 c0 a8 65 0e c0 a8   .P.=.... ^...e...
0020  65 ff 04 05 47 46 00 3c  0c 0d 0d 01 8e 00 c0 a8   e...GF.< ........
0030  65 0e 01 00 00 00 de ef  df 4f 08 ae c5 10 02 00   e....... .O......
0040  00 00 01 01 00 00 5c b5  4a 4e 04 00 00 00 5d 60   ......\. JN....]`
0050  e0 4f dd ef df 4f d9 38  7a 4e a5 3c 6c 4f         .O...O.8 zN.<lO
```

Let's look into the packet (3)

```
  243 19.53770100( 172.16.192.10   172.16.192.30   Modbus/TCP   78   Query: Trans:   0; Unit:   1, Func:   3: Read Holding Registers
  245 19.78689100( 172.16.192.30   172.16.192.10   Modbus/TCP   87 Response: Trans:   0; Unit:   1, Func:   3: Read Holding Registers

⊞ Frame 245: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
⊞ Ethernet II, Src: Raspberr_95:7b:ac (b8:27:eb:95:7b:ac), Dst: Vmware_b4:34:14 (00:0c:29:b4:34:14)
⊞ Internet Protocol Version 4, Src: 172.16.192.30 (172.16.192.30), Dst: 172.16.192.10 (172.16.192.10)
⊞ Transmission Control Protocol, Src Port: asa-appl-proto (502), Dst Port: 54034 (54034), Seq: 1544, Ack: 961, Len: 21
⊟ Modbus/TCP
    Transaction Identifier: 0
    Protocol Identifier: 0
    Length: 15
    Unit Identifier: 1
⊟ Modbus
    Function Code: Read Holding Registers (3)
    Byte Count: 12
    Register 0 (UINT16): 1
    Register 1 (UINT16): 0
    Register 2 (UINT16): 0
    Register 3 (UINT16): 1
    Register 4 (UINT16): 0
    Register 5 (UINT16): 0

0000  00 0c 29 b4 34 14 b8 27  eb 95 7b ac 08 00 45 00   ..).4..' ..{...E.
0010  00 49 85 4a 40 00 40 06  dd 1a ac 10 c0 1e ac 10   .I.J@.@. ........
0020  c0 0a 01 f6 d3 12 a9 b8  fe bd 85 ec 84 14 80 18   ........ ........
0030  01 c5 12 d3 00 00 01 01  08 0a 00 00 82 34 00 05   ........ .....4..
0040  70 f2 00 00 00 00 00 0f  01 03 0c 00 01 00 00 00   p....... ........
0050  00 00 01 00 00 00 00                              .......
```

Let's look into the packet (4)

No.	Time	Source	Destination	Protocol	Length	Info
42	79.717…	192.168.1.117	10.181.1.18	BACnet-APDU	59	Confirmed-REQ readProperty[12] device,1600 object-name
43	79.805…	10.181.1.18	192.168.1.117	BACnet-APDU	65	Complex-ACK readProperty[12] device,1600 object-name
44	87.170…	10.181.1.18	192.168.1.117	BACnet-APDU	127	Unconfirmed-REQ unconfirmedPrivateTransfer V=8 SN=1

```
▷ {[3]
▷ Abstract Type: 50.470615 (Real)
▷ }[3]
▷ Context Value (as 3 DATA octets)
▷ Context Value (as 4 DATA octets)
▷ Context Value (as 1 DATA octets)
▷ {[3]
▷ Abstract Type: 73.963737 (Real)
▷ }[3]
▷ Context Value (as 3 DATA octets)
▷ Context Value (as 4 DATA octets)
▷ Context Value (as 1 DATA octets)
▷ {[3]
▷ Abstract Type: 1087.412354 (Real)
```

Relative humidity

Temperature (F)

???

```
0020  01 75 ba c0 ba c0 00 5d  00 00 81 04 00 55 0a b5   .u.....] .....U..
0030  01 12 ba c0 01 00 10 04  09 08 19 01 2e 0c 02 00   ........ ........
0040  06 40 19 00 29 00 3e 0c  00 80 00 02 19 55 3e 44   .@..).>. .....U>D
0050  42 49 e1 e9 3f 5b 04 00  00 0c 00 80 00 01 19 55   BI..?[.. .......U
0060  3e 44 42 93 ed 6f 3f 5b  04 00 00 0c 00 80 00 05   >DB..o?[ ........
0070  19 55 3e 44 44 87 ed 32  3f 5b 04 00 00 3f 2f      .U>DD..2 ?[...?/
```

Let's look into the packet (5)

```
15:15:13.170324 172.21.21.1     172.21.21.77    Modbus/TCP   66    5487   Query: Trans:  6685; Unit:   1, Func:   3: Read Holding Registers
15:15:13.171773 172.21.21.77    172.21.21.1     Modbus/TCP   239        Response: Trans:  6685; Unit:   1, Func:   3: Read Holding Registers
15:15:13.185097 172.21.21.1     172.21.21.77    Modbus/TCP   66    6000   Query: Trans:  6686; Unit:   1, Func:   3: Read Holding Registers
15:15:13.186750 172.21.21.77    172.21.21.1     Modbus/TCP   179        Response: Trans:  6686; Unit:   1, Func:   3: Read Holding Registers
```

```
Register 5511 (Modicon Float): 0.000000
Register 5513 (Modicon Float): 0.000000
Register 5515 (Modicon Float): 71185.000000
Register 5517 (Modicon Float): 71185.000000
Register 5519 (Modicon Float): 8.525818
Register 5521 (Modicon Float): 8.694681
Register 5523 (Modicon Float): 1234.900635
Register 5525 (Modicon Float): 3.696970
Register 5527 (Modicon Float): 0.000000
```

```
0000  b8 ca 3a d5 a6 ff 00 50  c2 fe 50 42 08 00 45 00   ..:....P ..PB..E.
0010  00 e1 9f fe 00 00 ff 06  98 9f ac 15 15 4d ac 15   ........ .....M..
0020  15 01 01 f6 f2 35 00 bf  7b c3 2d cf b4 2a 50 18   .....5.. {.-..*P.
0030  1c ac 11 88 00 00 1a 1d  00 00 00 b3 01 03 b0 01   ........ ........
0040  ad 00 00 00 00 07 00 81  d0 01 59 00 00 06 00 81   ........ ..Y.....
0050  d1 01 59 00 00 06 00 2e  60 01 59 00 00 08 00 61   ..Y..... `.Y....a
0060  af 01 59 00 00 09 00 61  af 01 59 00 00 00 00 58   ..Y....a ..Y....X
0070  ea 00 00 00 28 00 00 08  80 47 8b 08 80 47 8b 69   ....(... .G...G.i
0080  c0 41 08 1d 6a 41 0b 5c  d2 44 9a 9b 26 40 6c 00   .A..jA.\ .D..&@l.
0090  00 00 00 01 2c 43 27 00  00 00 00 00 00 00 00 00   ....,C'. ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

https://ask.wireshark.org/questions/59670/extract-particular-register-from-series-of-modbus-packets

Let's look into the packet (6)

```
 434 1.070135   10.85.64.50    10.21.81.252   DNP 3.0   162 from 16 to 1024, len=255, Unconfirmed User Data, TL fragment 23 [TCP segment of a reassembled PDU]
 553 1.131345   10.85.64.50    10.21.81.252   DNP 3.0   112 from 16 to 1024, Response
 740 1.447104   10.21.81.252   10.85.64.50    DNP 3.0    78 from 1024 to 16, Read, Internal Indications
 749 1.510921   10.85.64.50    10.21.81.252   DNP 3.0    75 from 16 to 1024, Response
 777 1.844267   10.21.81.252   10.85.64.50    DNP 3.0    78 from 1024 to 16, Read, Internal Indications
 785 1.908871   10.85.64.50    10.21.81.252   DNP 3.0    75 from 16 to 1024, Response
1199 2.219736   10.21.81.252   10.85.64.50    DNP 3.0    78 from 1024 to 16, Read, Internal Indications
1211 2.283874   10.85.64.50    10.21.81.252   DNP 3.0    75 from 16 to 1024, Response
1269 2.594731   10.21.81.252   10.85.64.50    DNP 3.0    76 from 1024 to 16, Read, Class 0
1560 2.961068   10.85.64.50    10.21.81.252   DNP 3.0   162 from 16 to 1024, len=255, Unconfirmed User Data, TL fragment 28 [TCP segment of a reassembled PDU]
1571 3.022307   10.85.64.50    10.21.81.252   DNP 3.0   112 from 16 to 1024, Response
```

```
▷ Object(s): Binary Input With Status (Obj:01, Var:02) (0x0102), 40 points
◢ Object(s): 16-Bit Analog Input (Obj:30, Var:02) (0x1e02), 70 points
  ▷ Qualifier Field, Prefix: None, Range: 8-bit Start and Stop Indices
  ◢ Point Number 0 (Quality: Online), Value: 1678
      [Point Index: 0]
    ▷ Quality: Online
      Value (16 bit): 1678
  ▷ Point Number 1 (Quality: Online), Value: 1358
  ▷ Point Number 2 (Quality: Online), Value: 1760
  ▷ Point Number 3 (Quality: Online), Value: 1677
  ▷ Point Number 4 (Quality: Online), Value: 1629
  ▷ Point Number 5 (Quality: Online), Value: 1803
  ▷ Point Number 6 (Quality: Online), Value: 74
  ▷ Point Number 7 (Quality: Online), Value: 103
  ▷ Point Number 8 (Quality: Online), Value: 25
```

```
0030  81 1e 02 00 00 45 01 8e  06 01 4e 05 01 e0 06 01   .....E.. ..N.....
0040  8d 06 01 5d 06 01 0b 07  01 4a 00 01 67 00 01 19   ...]....  .J..g...
0050  00 01 0f 00 01 f1 00 01  74 00 01 da 00 01 05 01   ........  t.......
0060  01 f3 00 01 fb 00 01 b7  00 01 b6 00 01 b6 00 01   ........  ........
0070  b7 00 01 b9 00 01 b6 00  01 01 80 01 01 80 01 01   ........  ........
0080  80 01 01 80 01 01 80 01  01 80 01 0e 0e 01 0f 0e   ........  ........
0090  01 0f 0e 01 11 0e 01 0f  0e 01 0f 0e 21 5b 12 21   ........  ....![.!
```

Let's look into the packet (7)

42	22.216012	192.168.0.100	192.168.0.2	Modbus/TCP	66	Query: Trans: 2; Unit: 1, Func: 6: Write Single Register
43	22.223304	192.168.0.2	192.168.0.100	TCP	60	502 → 15425 [ACK] Seq=90 Ack=85 Win=11680 Len=0
44	22.230517	192.168.0.2	192.168.0.100	Modbus/TCP	66	Response: Trans: 2; Unit: 1, Func: 6: Write Single Register
45	22.431041	192.168.0.100	192.168.0.2	TCP	54	15425 → 502 [ACK] Seq=85 Ack=102 Win=65419 Len=0
46	28.010511	192.168.0.100	192.168.0.2	Modbus/TCP	66	Query: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
47	28.013147	192.168.0.2	192.168.0.100	TCP	60	502 → 15425 [ACK] Seq=102 Ack=97 Win=11668 Len=0
48	28.025390	192.168.0.2	192.168.0.100	Modbus/TCP	83	Response: Trans: 2; Unit: 1, Func: 3: Read Holding Registers
49	28.230019	192.168.0.100	192.168.0.2	TCP	54	15425 → 502 [ACK] Seq=97 Ack=131 Win=65390 Len=0

```
▷ Frame 48: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
▷ Ethernet II, Src: PhoenixC_8c:36:75 (00:a0:45:8c:36:75), Dst: WistronI_a4:f5:3a (3c:97:0e:a4:f5:3a)
▷ Internet Protocol Version 4, Src: 192.168.0.2, Dst: 192.168.0.100
▷ Transmission Control Protocol, Src Port: 502, Dst Port: 15425, Seq: 102, Ack: 97, Len: 29
▷ Modbus/TCP
◢ Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    [Request Frame: 46]
    Byte Count: 20
    Register 0 (UINT16): 104
    Register 1 (UINT16): 97
    Register 2 (UINT16): 99
    Register 3 (UINT16): 107
    Register 4 (UINT16): 101
    Register 5 (UINT16): 100
    Register 6 (UINT16): 0
    Register 7 (UINT16): 0
    Register 8 (UINT16): 0
    Register 9 (UINT16): 0
```

```
0000  3c 97 0e a4 f5 3a 00 a0  45 8c 36 75 08 00 45 00   <....:.. E.6u..E.
0010  00 45 00 11 00 00 40 06  f8 eb c0 a8 00 02 c0 a8   .E....@. ........
0020  00 64 01 f6 3c 41 00 44  7e da e2 88 bc c9 50 18   .d..<A.D ~.....P.
0030  2d a0 2e 92 00 00 00 02  00 00 00 17 01 03 14 00   -....... ........
0040  68 00 61 00 63 00 6b 00  65 00 64 00 00 00 00 00   h.a.c.k. e.d.....
```

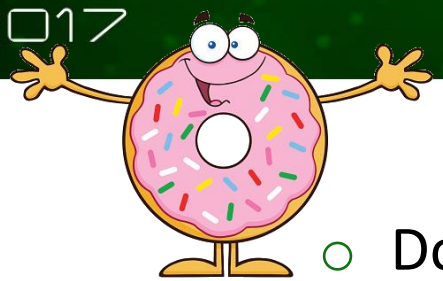
Let's look into the packet (8)

No.	Time	Source	Destination	Protocol	Length	Info
7	28.475…	192.168.224.1	192.168.224.90	DNP 3.0	334	from 36541 to 65432, len=255, Unconfirmed User Data, TL fragment 1
8	28.574…	192.168.224.1	192.168.224.90	DNP 3.0	122	from 36541 to 65432, Response
9	37.322…	192.168.224.90	192.168.224.1	DNP 3.0	69	from 65432 to 65532, Read, Class 0123

```
▷ Point Number 20, Value: 1
▷ Point Number 21, Value: 1
▷ Point Number 22, Value: 0
◢ Object(s): 16-Bit Analog Input Without Flag (Obj:30, Var:04) (0x1e04), 6 points
  ▷ Qualifier Field, Prefix: None, Range: 8-bit Start and Stop indices
  ▷ [Number of Items: 6]
  ▷ Point Number 0, Value: 0
  ▷ Point Number 1, Value: 56
  ▷ Point Number 2, Value: 17
  ▷ Point Number 3, Value: 0
```

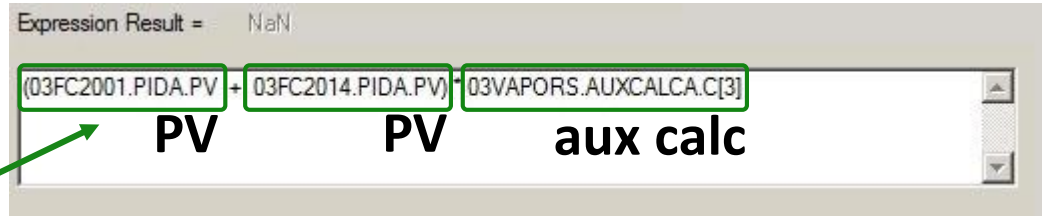
```
01a0  14 81 d0 23 4a 48 3a 01  14 01 e9 46 4a 48 3a 01   ...#JH:. ...FJH:.
01b0  14 81 bd 4e 4a 48 3a 01  14 01 9a 56 4a 48 3a 01   ...NJH:. ...VJH:.
01c0  14 81 6d 62 4a 48 3a 01  14 01 35 07 4b 48 3a 01   ..mbJH:. ..5.KH:.
01d0  14 81 17 0b 4b 48 3a 01  14 01 55 36 4b 48 3a 01   ....KH:. ..U6KH:.
01e0  14 81 30 3a 4b 48 3a 01  14 01 a8 b9 4e 48 3a 01   ..0:KH:. ....NH:.
01f0  14 81 87 bd 4e 48 3a 01  01 81 d9 23 50 48 3a 01   ....NH:. ...#PH:.
0200  09 81 60 f4 70 48 3a 01  01 01 00 00 16 83 43 30   ..`.pH:. ......C0
0210  1e 04 00 00 05 00 00 38  00 11 00 00 00 00 00 00   .......8 ........
0220  00 0a 02 00 00 08 01 01  01 01 01 01 01 01 01      ........ .......
```

Who can guess it best?



17

- Donuts? (Ok, it was a joke)
- Can be
  - Direct measurement
  - Result of computation

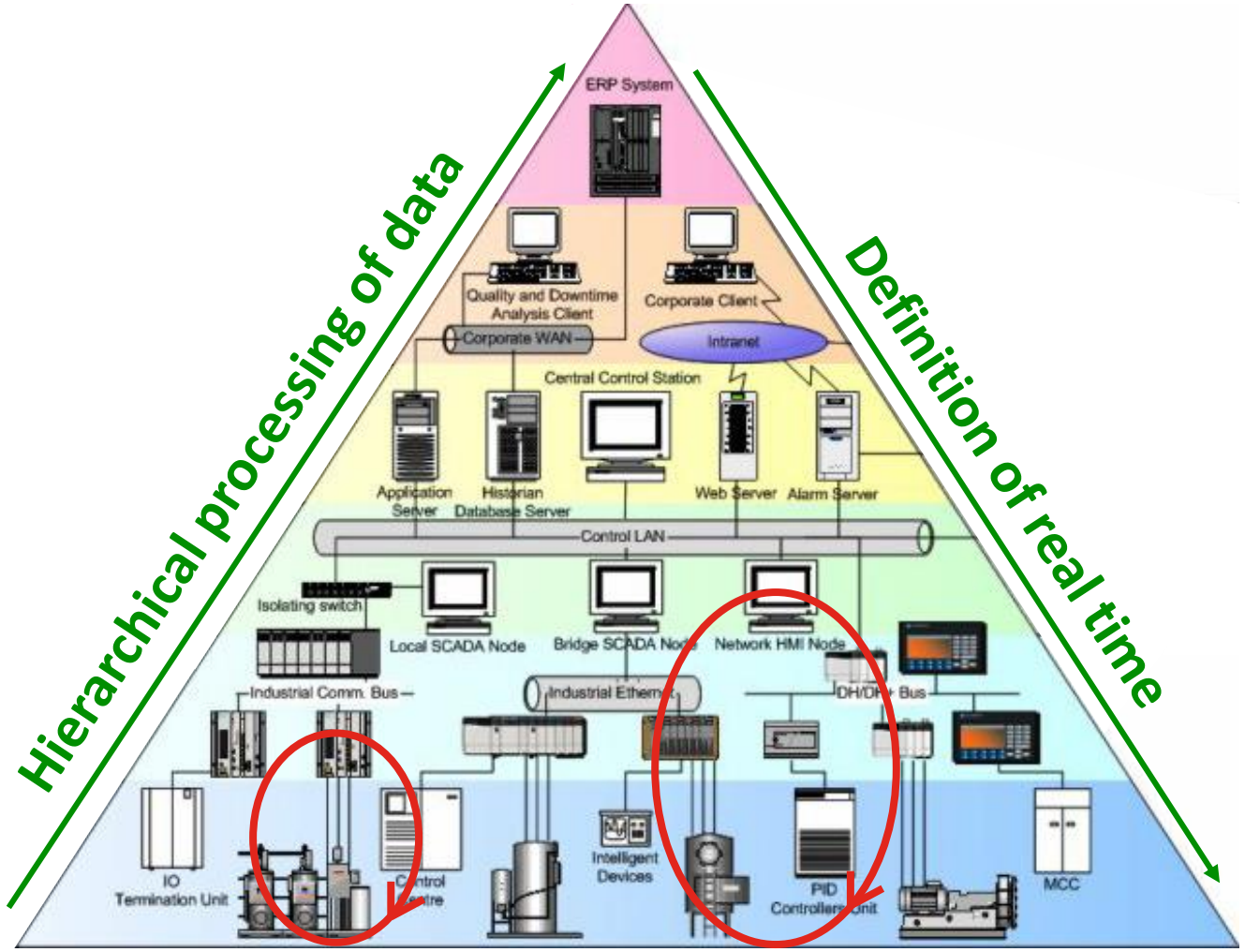


- Bit counts/%/EU **EU -> Engineering Units**
- Celsius/Fahrenheit
- Centimeters/meters/miles/light years
- Pa/kPa/mPa/Psia/Psig/Atm/Bar
- Kgh/m^3h/nm^3h/scmh/kscmh
- Keep guessing….

New Information to Build New Assumptions

Configuration of a Single point

Purdue reference architecture

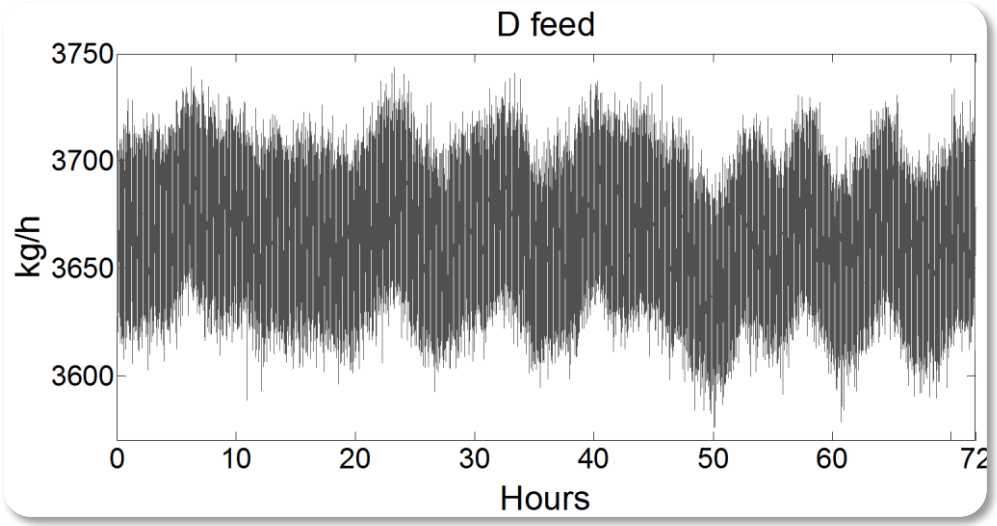


Operates on raw data

Operates on information

http://krakenautomation.com/images/KrakenPyramid.jpg

Raw measurement



- **Raw sensory data rarely can be used directly.** The electrical output of a sensing element is usually small in value and has non-idealities such as offset, sensitivity errors, nonlinearities, noise, etc.
- Raw transducer output is subjected to signal conditioning such as amplification, filtering, range matching, etc.

Point configuration

Everything what can be measured or set is called POINT

+/-10 V dc
0-10 V dc
0-5 V dc
4-20 mA

1	0	1	0	1	1	1	0	1	0	0	1	**0-4095**	I/O card with ADC
1	0	1	0	1	1	1	0	1	0	0	1		I/O card with ADC
1	0	1	0	1	1	1	0	1	0	0	1		I/O card with ADC
1	0	1	0	1	1	1	0	1	0	0	1		I/O card with ADC

4-20 mA,


0-32 m^3/h


40-100 psi


0-70 m^3/h


0-8000 kg/h

Sensor calibration:
e.g. measuring from **0** to **32** m^3/h

Low and High limits:
LO 10 m^3/h
HI 25 m^3/h

12 bit ADC resolution
(defines the quality of data translation)

Point configuration

Everything what can be measured or set is called POINT

+/-10 V dc
0-10 V dc
0-5 V dc
4-20 mA

1	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**
0	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1

3017 ← I/O card with ADC ← **14 mA,** —  20 m^3/h

1024 ← I/O card with ADC ← 8 mA —  55 psi

2048 ← I/O card with ADC ← 12 mA —  35 m^3/h

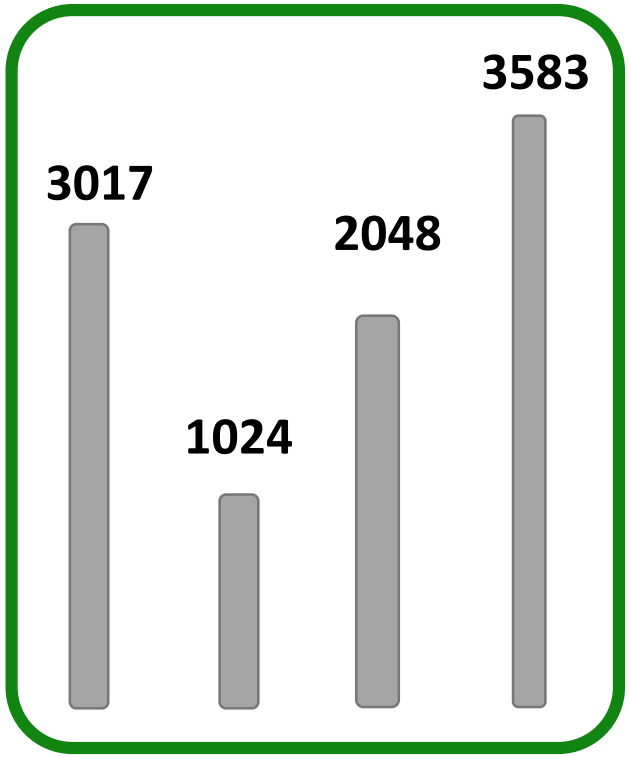
3583 ← I/O card with ADC ← 18 mA —  7000 kg/h

Sensor calibration:
e.g. measuring from 0 to 30 m^3/h

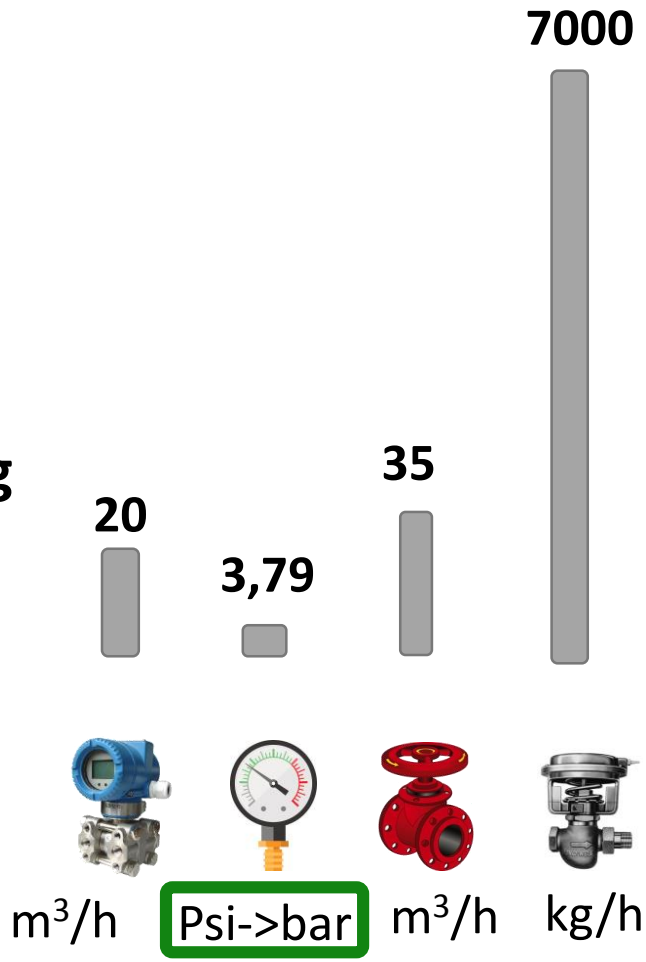
Low and High limits:
LO 10 m^3/h
HI 25 m^3/h

12 bit ADC resolution
(defines the quality of data translation)

Scaling of data into useful units



scaling

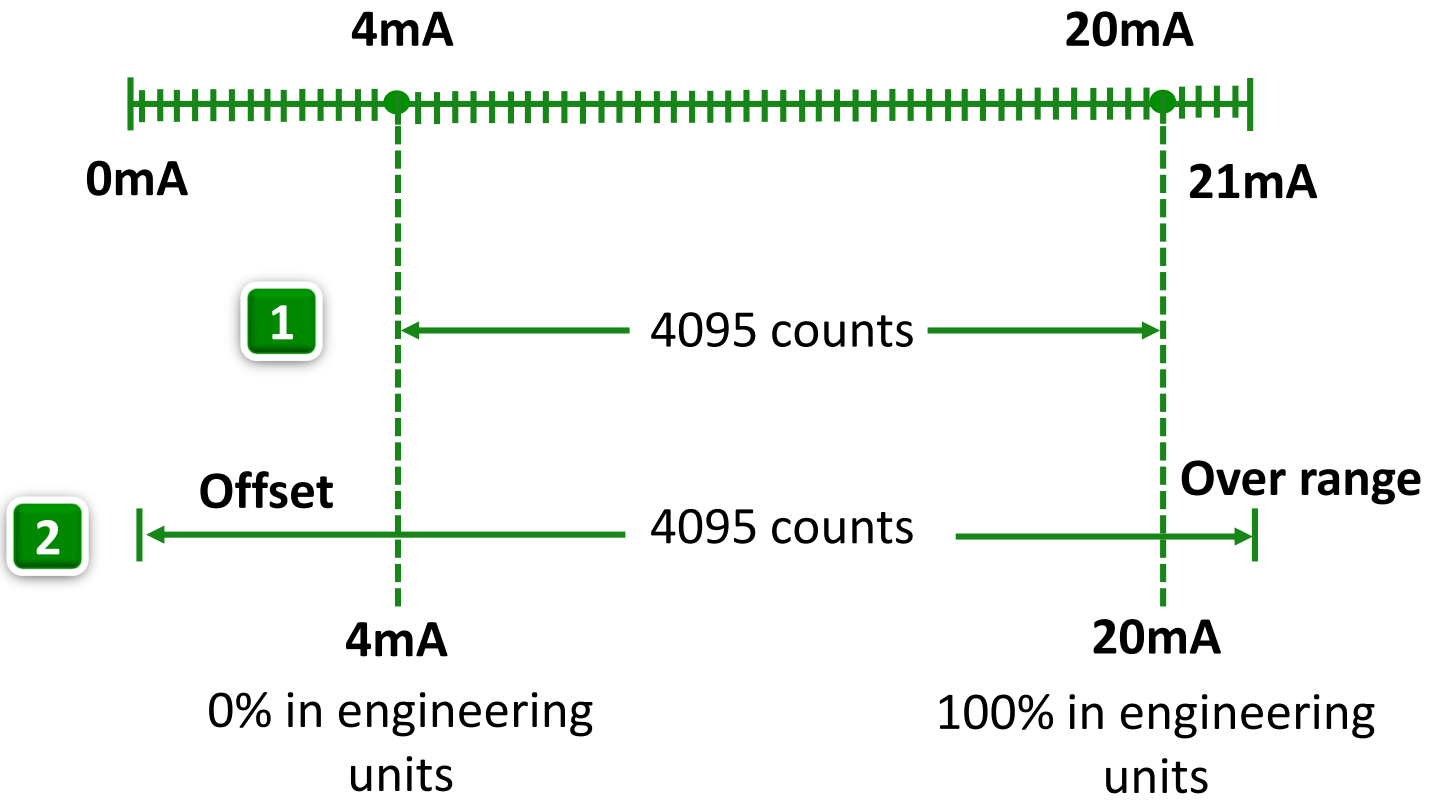


Raw counts are scaled into useful units, which could be different to different data users

Raw data

Engineering Units

Conversion of raw data into EU



Current	EU values
3 mA	-6,25%
4mA	0%
12mA	50%
20mA	100%
21mA	106,25%

Data scaling is case-specific

Getting point config right

http://control.com/thread/981642471

***I am working at a pump station trying to get it going**, I am a civil engineer not electrical so here's my question - The PLC guy is taking all the analog signals, take for example a pressure transmitter (0 to 150 psi range) and in his program uses some 0 to 4095 range to display the signal where as the telemetry guy uses 819 to 4095 to figure his signal so when the PLC get zero pressure he gets 0 but when the RTU gets zero pressure he gets 819 in some field in his program. Anyway you can see where this could lead to a problem if the signal goes to the RTU first and then the PLC or vice versa. SCADA reads everything from the RTU whereas the HMI at the site take everything from the PLC and things are not matching up. Someone gotta give but who's right? Where did they come up with 819 to 4095? That's my main question. Any help is appreciated. Thanks.*

0-150 psi range -> **4-20 mA**
12 bit ADC (0-4095)

0-4095 **819-4095**

- One engineer uses 819 counts offset to detect under-range, another does not
- Leads to inconsistencies in data readings across data path

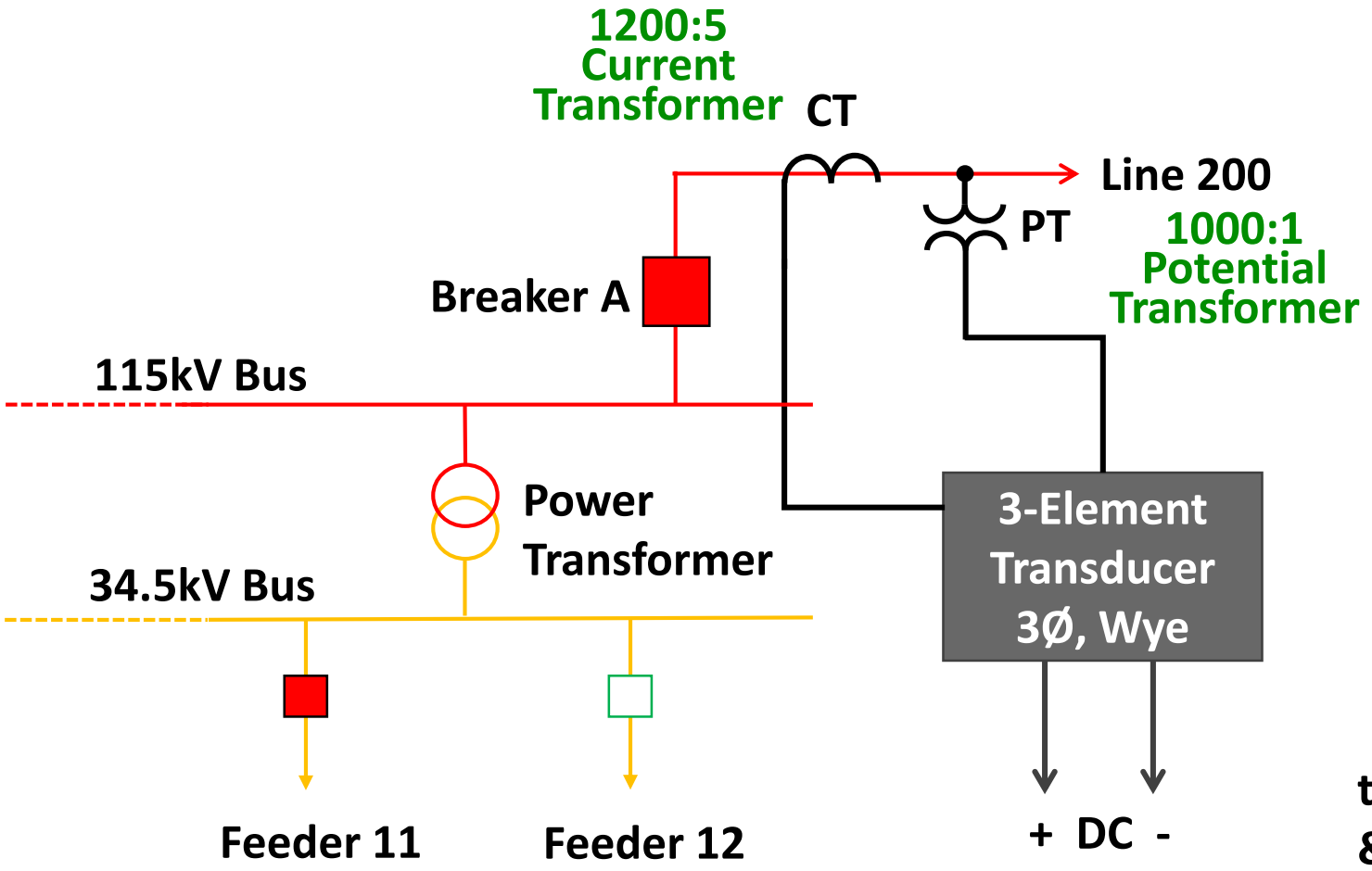
Learning More from Use Cases

Use Case1: Power Substation

Measuring power line

Chris Sistrunk at Power Substation

Measuring power line



How?

90 MW
114 kV
468 Amps

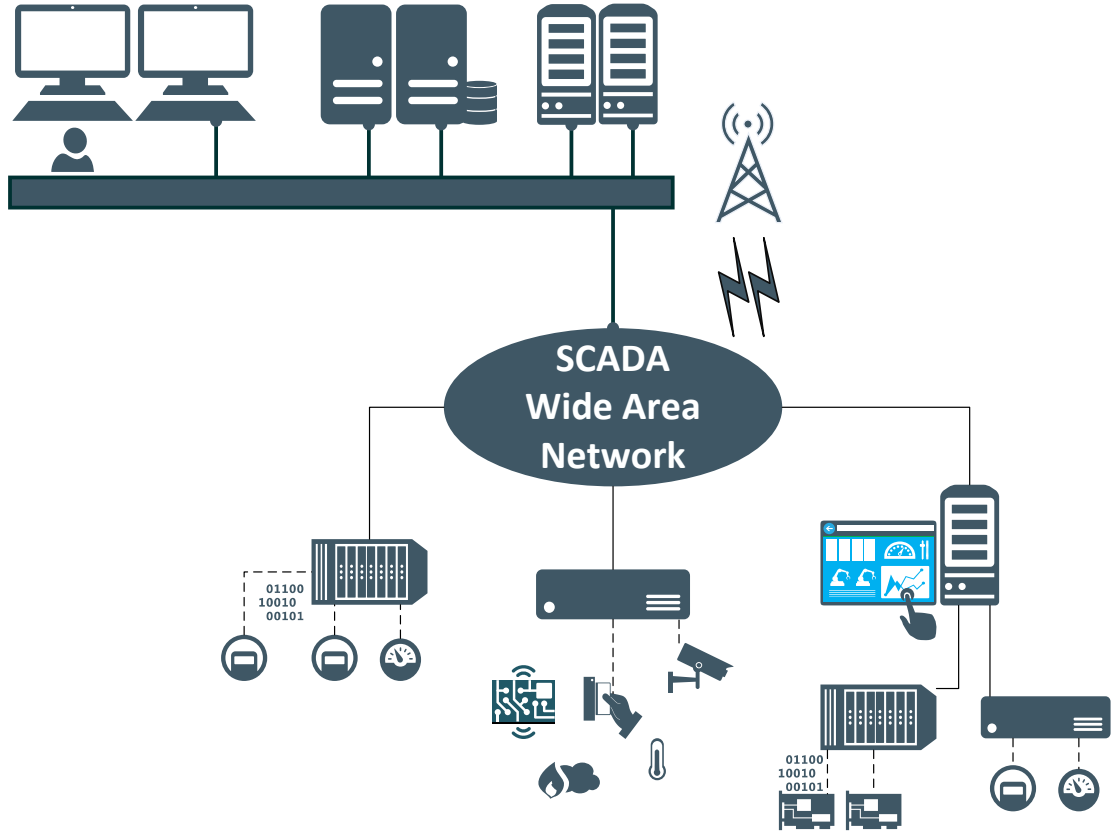
to Relays, Panel Meter,
& SCADA RTU, HMI

Properly select PT and CT ratio to allow some % of overload on the circuit, so the measurements will not top out at 100% when the actual values are higher.

Measuring power line

Level 5 – Enterprise Network	
Level 4 – IT Apps, Outage Mgmt, Billing	OMS
DMZ – Mirror Historian, Applications	HIS
Level 3 – EMS, Historian	EMS HIS
Level 2 – Front End, SCADA Master	FEP SCADA
Level 1 – Transducer, Meter, RTU	XDUCER RTU
Level 0 – CT, PT	

Data Flow



Power substation equipment

- Typically **multivendor**
- Non-homogeneous configuration requirements
- Decentralized configuration
- Requires careful integration
- Often (still) old equipment and networks with limited resources and bandwidth

Level 0

$$y_{SCADA} = m * x_{LINE} + b$$

x_{LINE} – initial (measured) value

m, b – scaling factor and offset for each time the data moves from one device to another

PT

CT

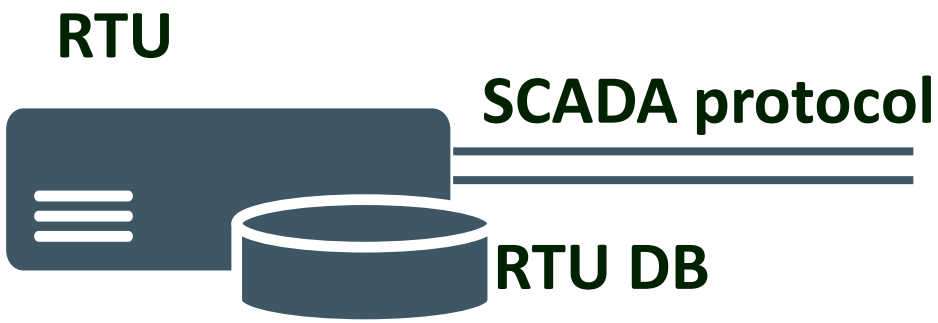
Transducers may be:
0 – 1mA
or
4 – 20mA
(which require an offset ***b***)

- MW Engineering Limit = (PT ratio) * (CT ratio) * (Transducer Multiplier) * (Line Connection Type) = (1200/5)(1000)(1500)(1)/1000000 = 300MW

- Transducer Output Range = 0 to +/-1mA → 0 – +/-300MW/mA scale

  If transducer output = 0.25mA, then 0.25*300 = 90 MW

Level 1



- RTU Analog input card (16-bit Analog to Digital Converter) 15 bits plus +/- sign bit
  -32768 to +32767 counts = -1mA to 1mA = 300MW/mA
  +90 MW = .25*32767 = **+8192 counts**
- RTU Database = same size → 90MW is stored as **+8192 bits** (+25% of db)
- SCADA Protocol has 12-bit bipolar analogs (-2048 to 2047 counts)
  SCADA protocol value MW = .25*2047 = **512 counts**

Level 2

- +512 bipolar counts from RTU to Front End Processor on a 12-bit protocol (0 – 4095)
  1 count = 300MW/2047 = 0.073242 MW per count unipolar
  (remember MW is a bipolar value)

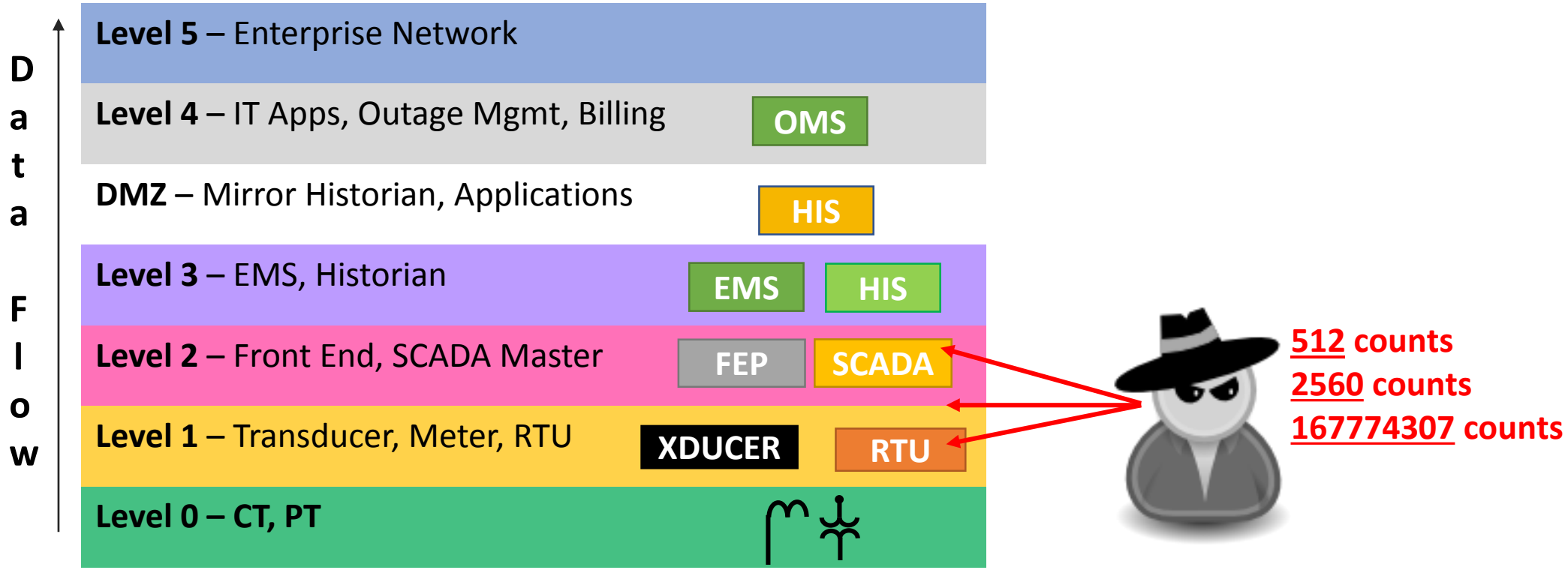
The FEP has to shift the bipolar value to a unipolar value to store it in the database!

- FEP database value = 512 incoming counts + offset of 2048 = **2560 counts**
  FEP database = 16 bits = 0 – 32767 counts
  2560 counts / 65535 counts = 0.039063 = 3.906309%

- SCADA database = 32 bits = 0 – 4294967295 counts
  3.906309% * 4294967295 = **167774307 counts**

AND SO ON….

Interpreting power data



Reverse engineering process data

Exploitation:
traditional IT hacking tools

Post-Exploitation:
engineering attack tools

Obtaining point configuration

- From the individual devices (e.g. RTU, FEP, DB, etc.)
  - May or may not be easy/rational thing to do
- From servers
- From individual config files on workstations

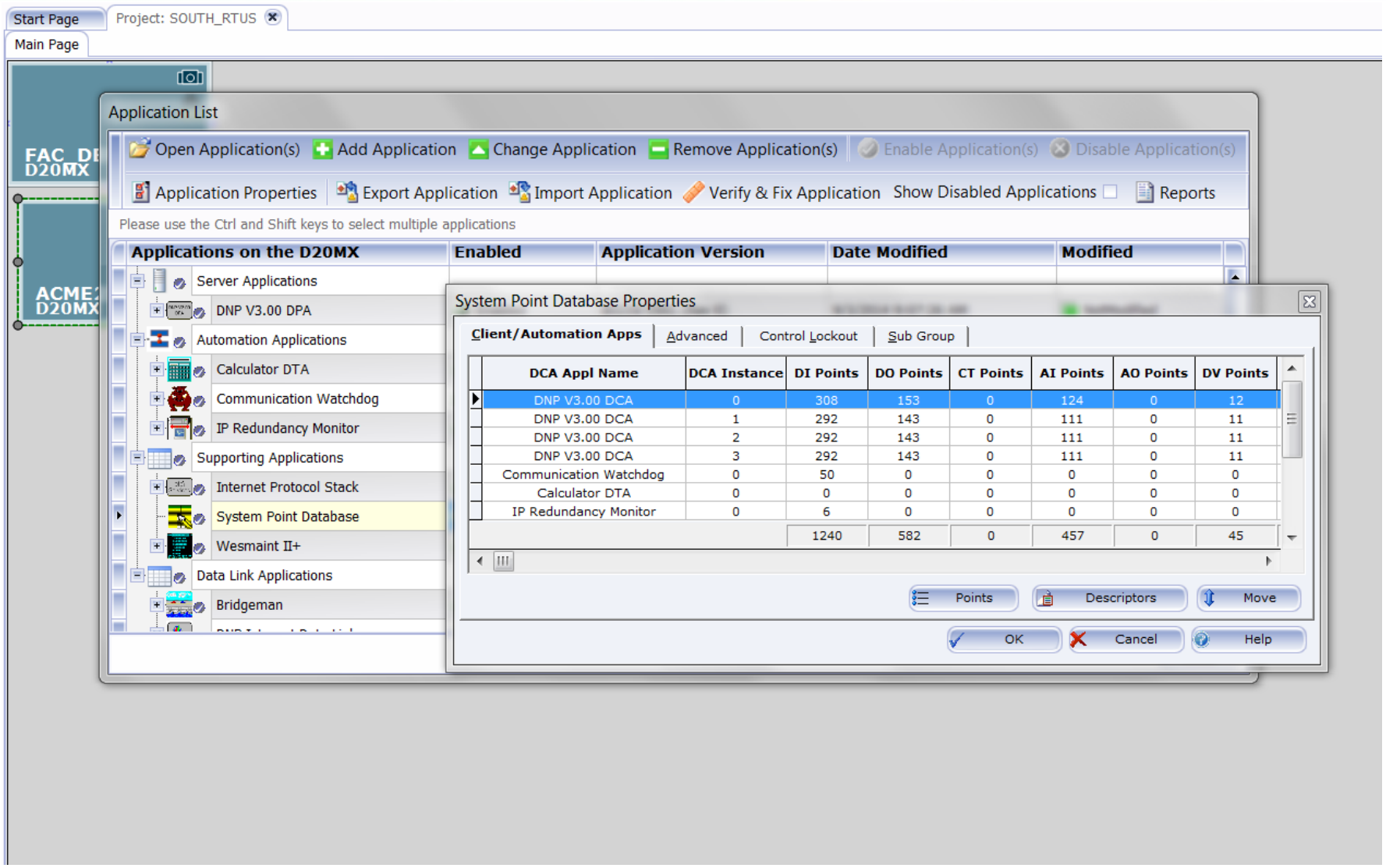
27th Chaos Communication Congress
We come in peace

Building Custom Disassemblers

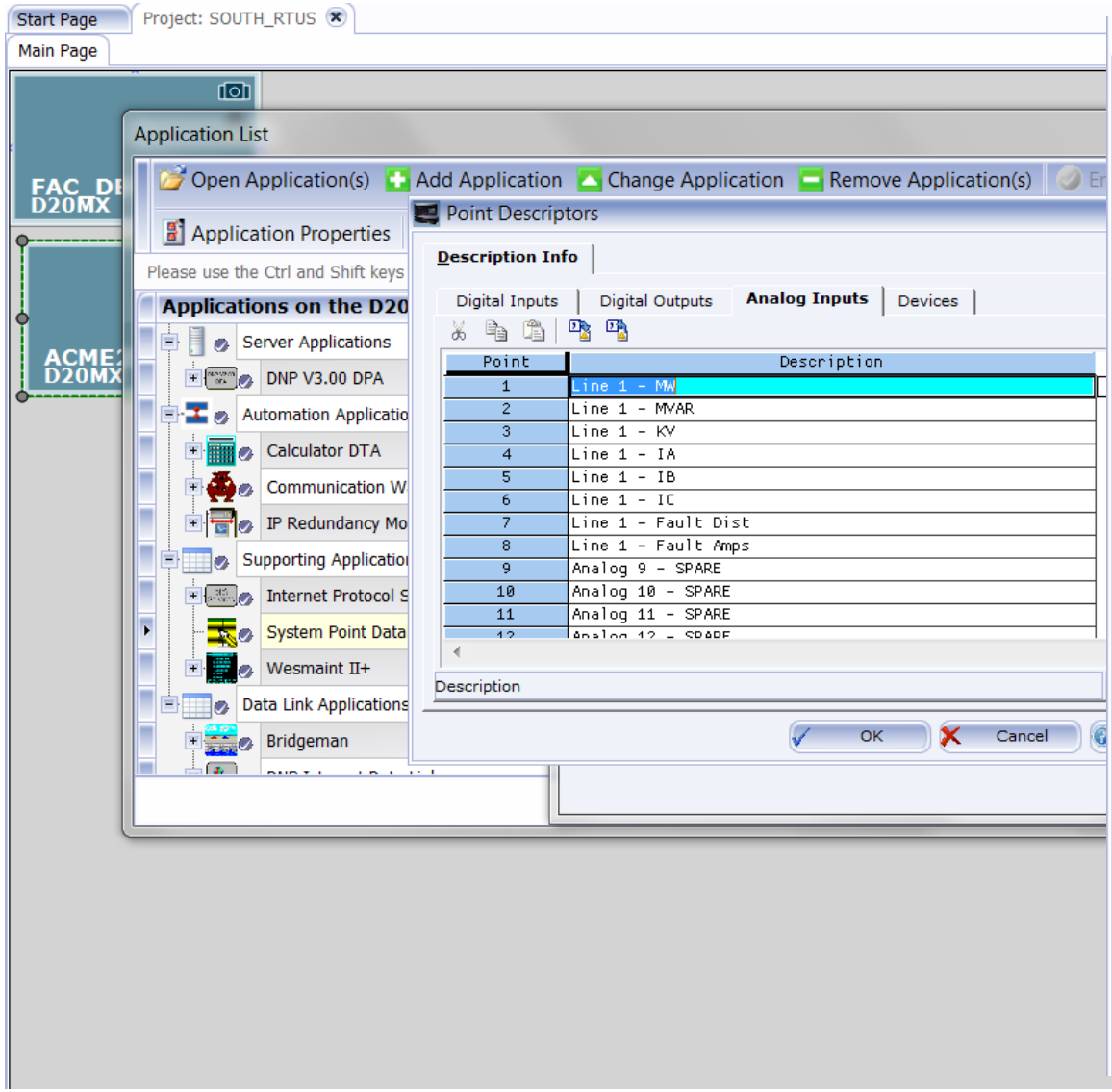
Instruction Set Reverse Engineering

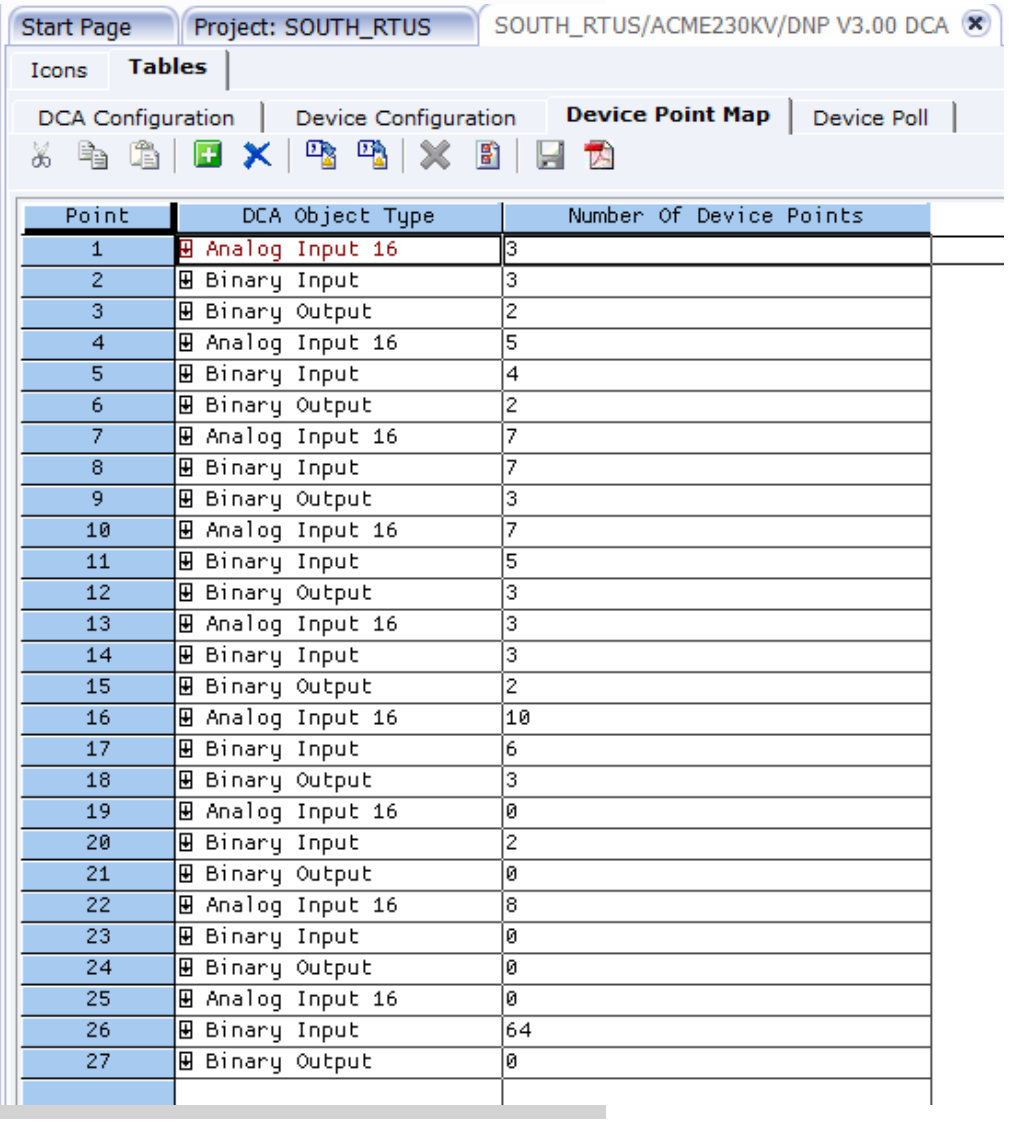
http://data.proidea.org.pl/confidence/9edycja/materialy/prezentacje/FX.pdf

With engineering applications



With engineering applications





Point	DCA Object Type	Number Of Device Points
1	Analog Input 16	3
2	Binary Input	3
3	Binary Output	2
4	Analog Input 16	5
5	Binary Input	4
6	Binary Output	2
7	Analog Input 16	7
8	Binary Input	7
9	Binary Output	3
10	Analog Input 16	7
11	Binary Input	5
12	Binary Output	3
13	Analog Input 16	3
14	Binary Input	3
15	Binary Output	2
16	Analog Input 16	10
17	Binary Input	6
18	Binary Output	3
19	Analog Input 16	0
20	Binary Input	2
21	Binary Output	0
22	Analog Input 16	8
23	Binary Input	0
24	Binary Output	0
25	Analog Input 16	0
26	Binary Input	64
27	Binary Output	0

Excel sheets of helpful engineers

Enter Engineering Limits	
360.00	MW
150.00	KV
1200.00	AMPS

Note: PT and CT ratios are chosen with the proper amount of engineering overrange
Example: 150kV will normally be used on a 115kV line (~30% overrange)
The next tab (Substation Transducer Calc) has many popular PT and CT ratios for convenience

Analog Meter Readings (0-1mA range)

Enter mA reading	MW	Enter mA reading	kV	Enter mA reading	AMPS
0.25	90.00	0.76	114.00	0.39	468.00

Digital SCADA Values (in the Substation RTU to the SCADA Host)

Enter MW	ADC	RTU Database	RTU Protocol Bipolar	Host Raw	Host Engr Limit	
	14-bit counts	16-bit counts	12-bit counts	12-bit counts	Eng Hi	360
90.00	4096	8192	512	2560	Eng Lo	-360

Enter kV	ADC	RTU Database	RTU Protocol Unipola	Host Raw	Host Engr Limit	
	14-bit counts	16-bit counts	12-bit counts	12-bit counts	Eng Hi	150
114.00	12451	24903	3112	3604	Eng Lo	0

Enter AMPS	ADC	RTU Database	RTU Protocol Bipolar	Host Raw	Host Engr Limit	
	14-bit counts	16-bit counts	12-bit counts	12-bit counts	Eng Hi	1200
468.00	6389	12779	798	2846	Eng Lo	-1200

The ADC	The RTU DB	This SCADA Protocol
has 14-bit	has 16-bit	has 12-bit analogs
resolution	analog register	Ex: Telegyr 8979
with a sign bit		Yes I wanted to show how difficult scaling could be with
		This happens in the real world...AHHH...why not ju

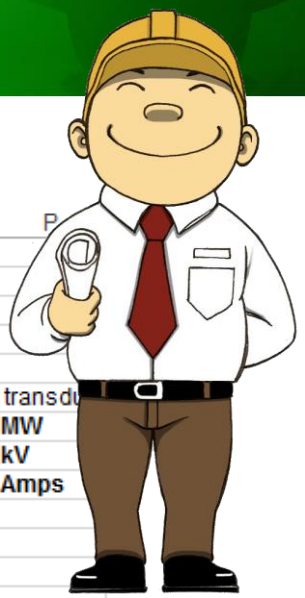
Chris Sistrunk

2.5 1500
3 1500

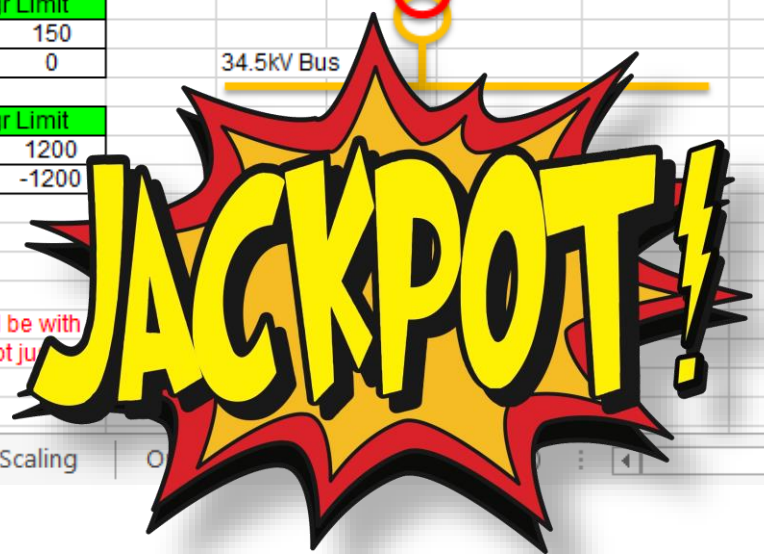
Simplified Substation One-line

1200:5 CT
Line 1
PT
1000:1
Phase-to-Phase
BKR A
115kV Bus
XFMR 1
34.5kV Bus

3-element transdu
90.00 MW
114.00 kV
468.00 Amps



THE HELPFUL ENGINEER



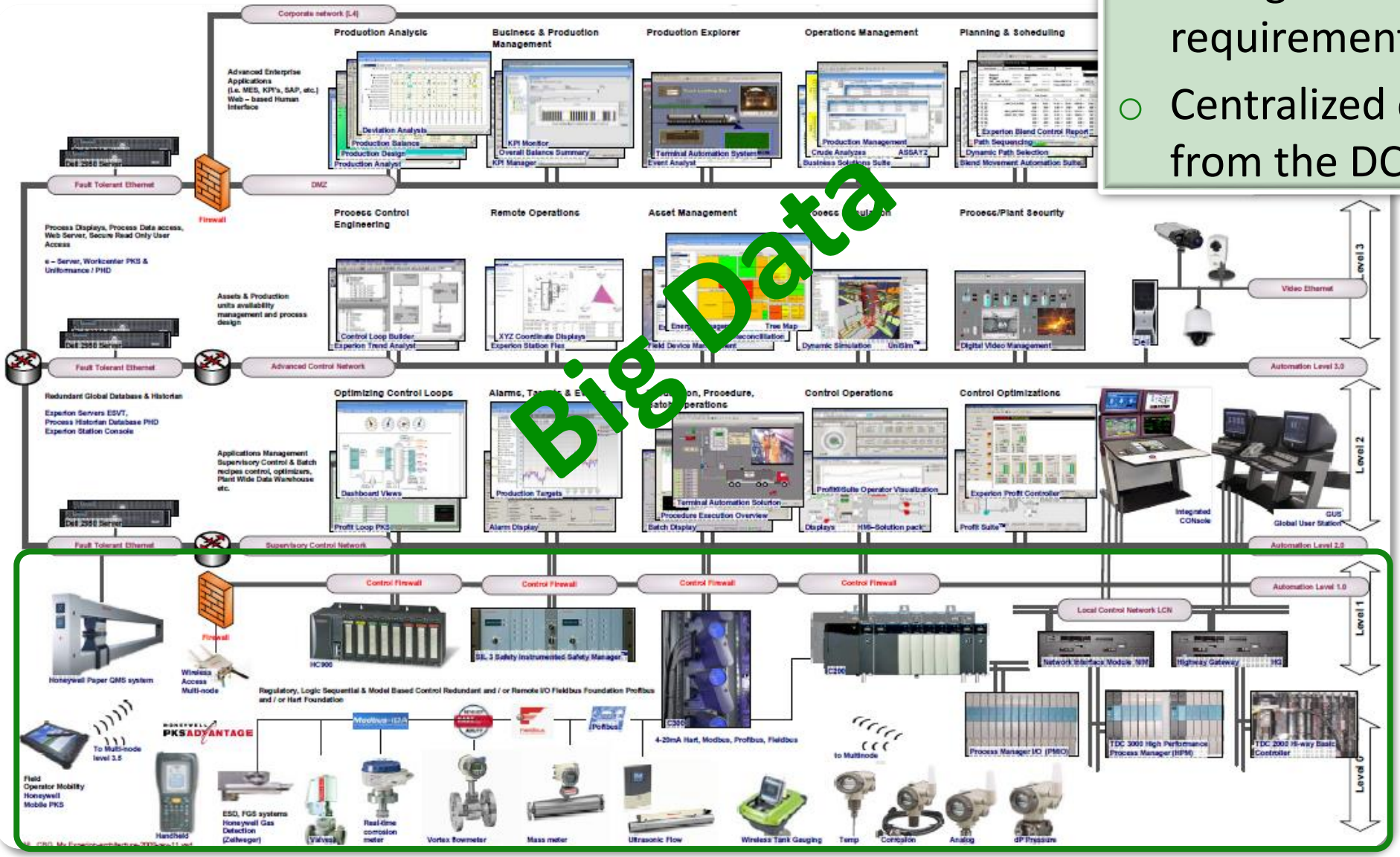
Example | Substation Transducer Calc | Subnet Solutions-Host Scaling | O

Learning More from Use Cases

Use Case2: Distributed Control System (DCS)

Typical architecture

- **Single vendor**
- Homogeneous configuration requirements
- Centralized configuration from the DCS server



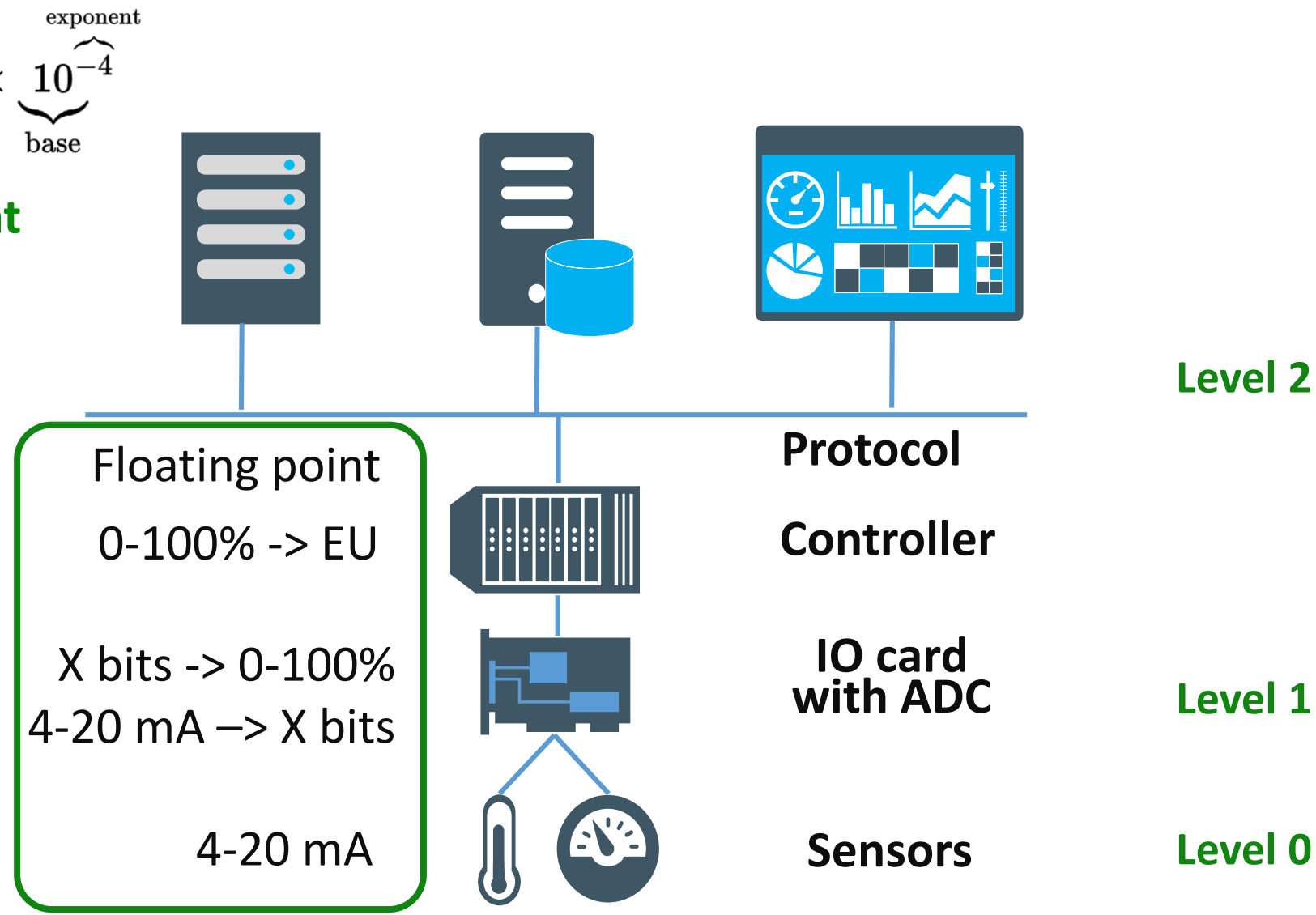
Typical data scaling

http://steve.hollasch.net/cgindex/coding/ieeefloat.html

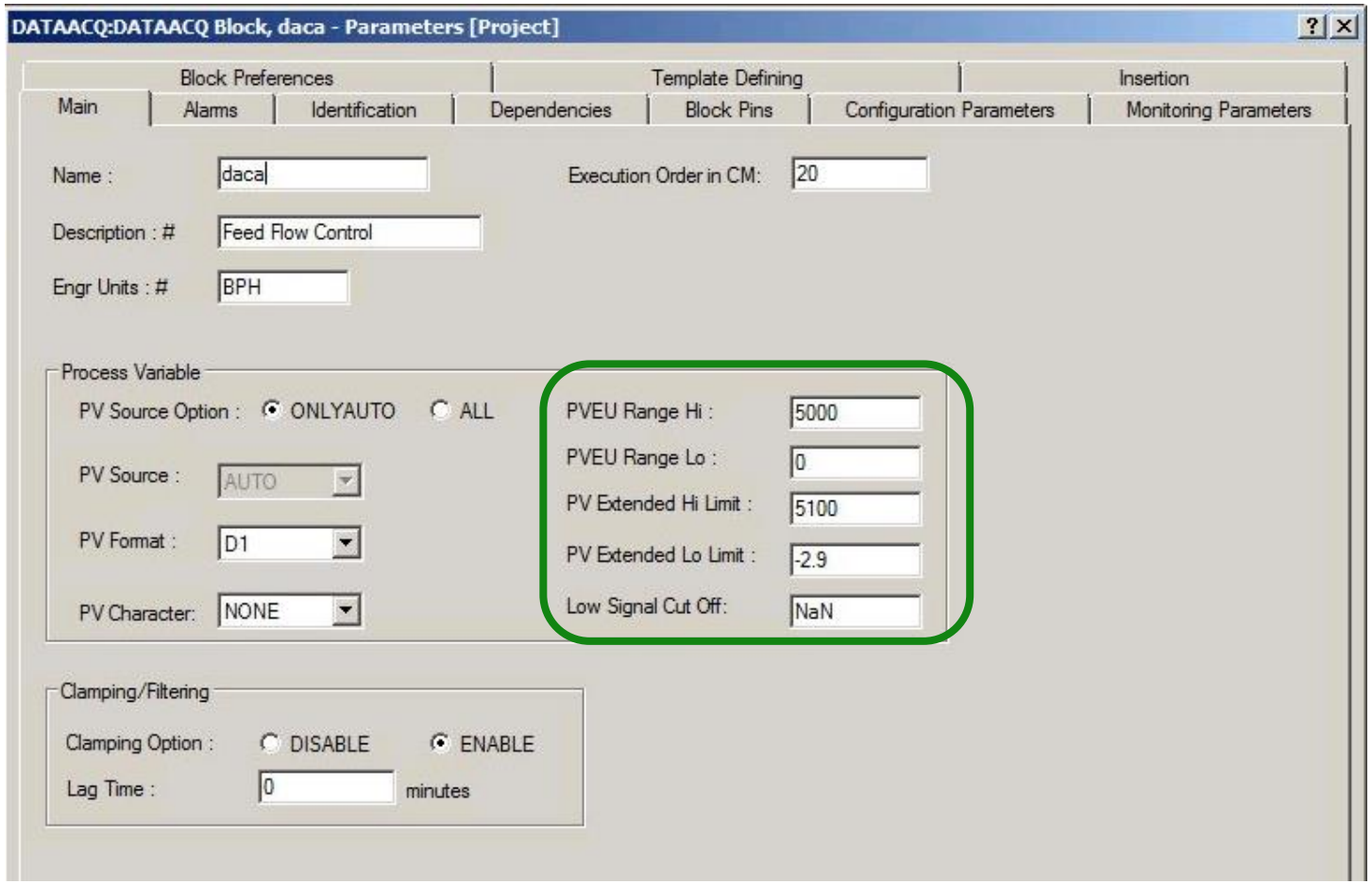
IEEE Standard 754
Floating Point Numbers

$$1.2345 = \underbrace{12345}_{\text{significand}} \times \underbrace{10^{\overbrace{-4}^{\text{exponent}}}}_{\text{base}}$$

Floating point

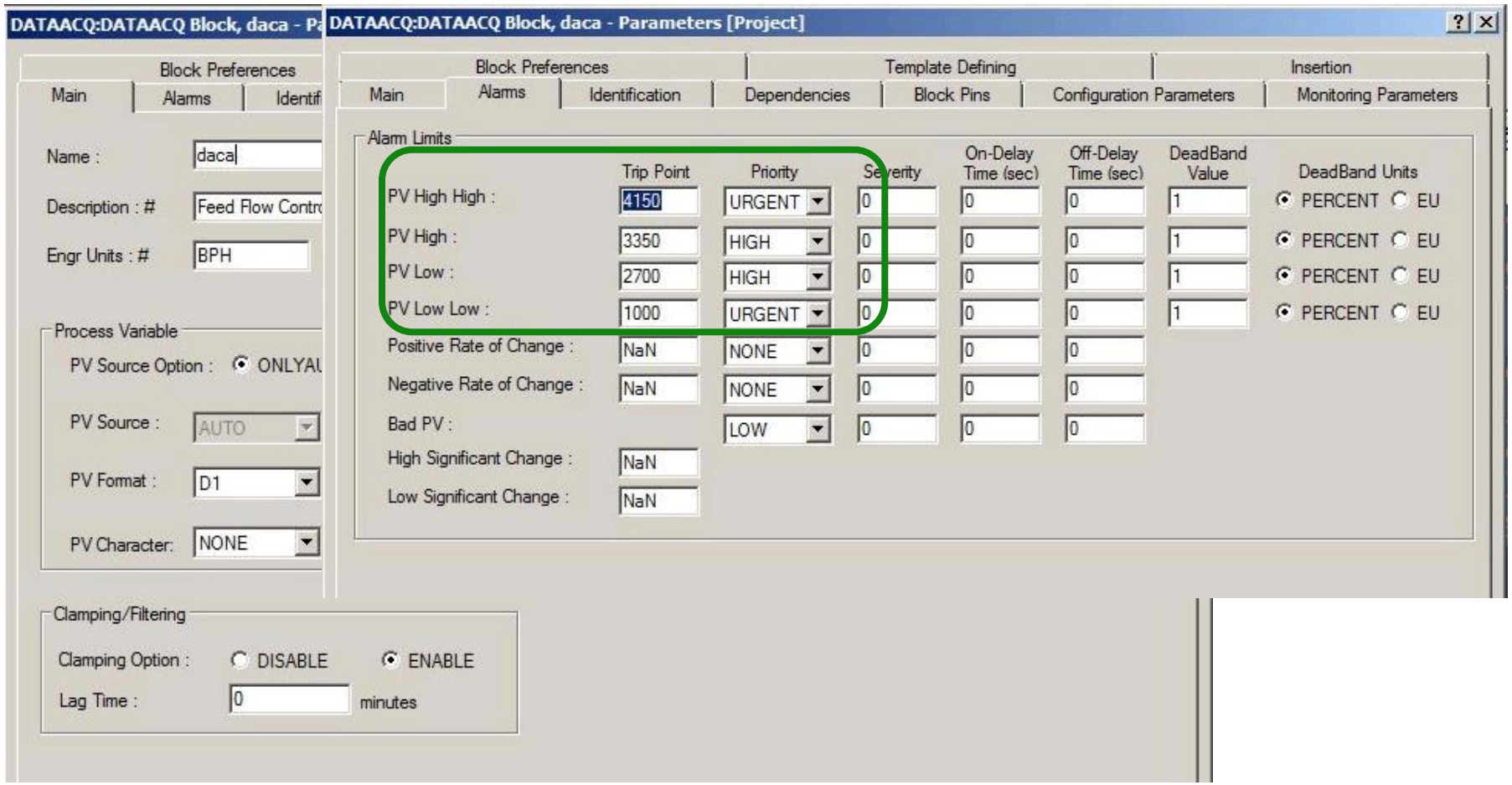


Point configuration



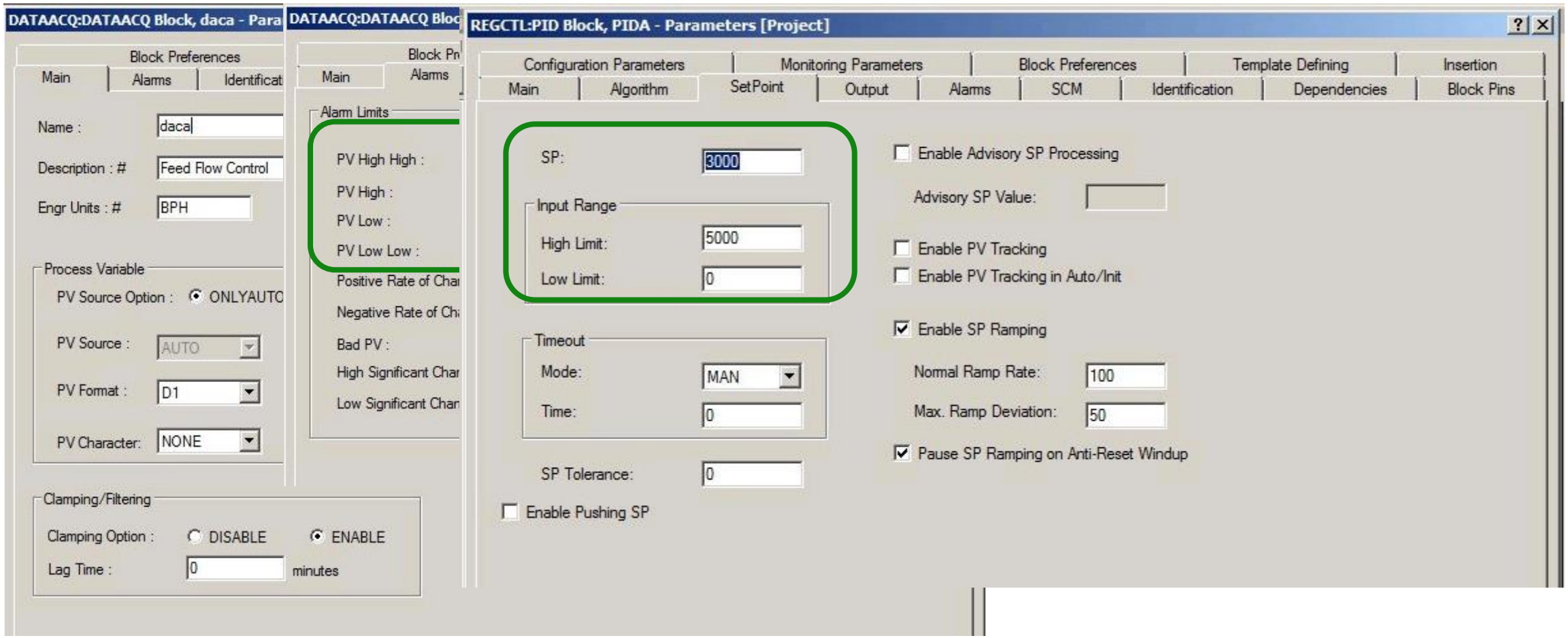
Point configuration is loaded into controller and stored on a DCS server

Point configuration



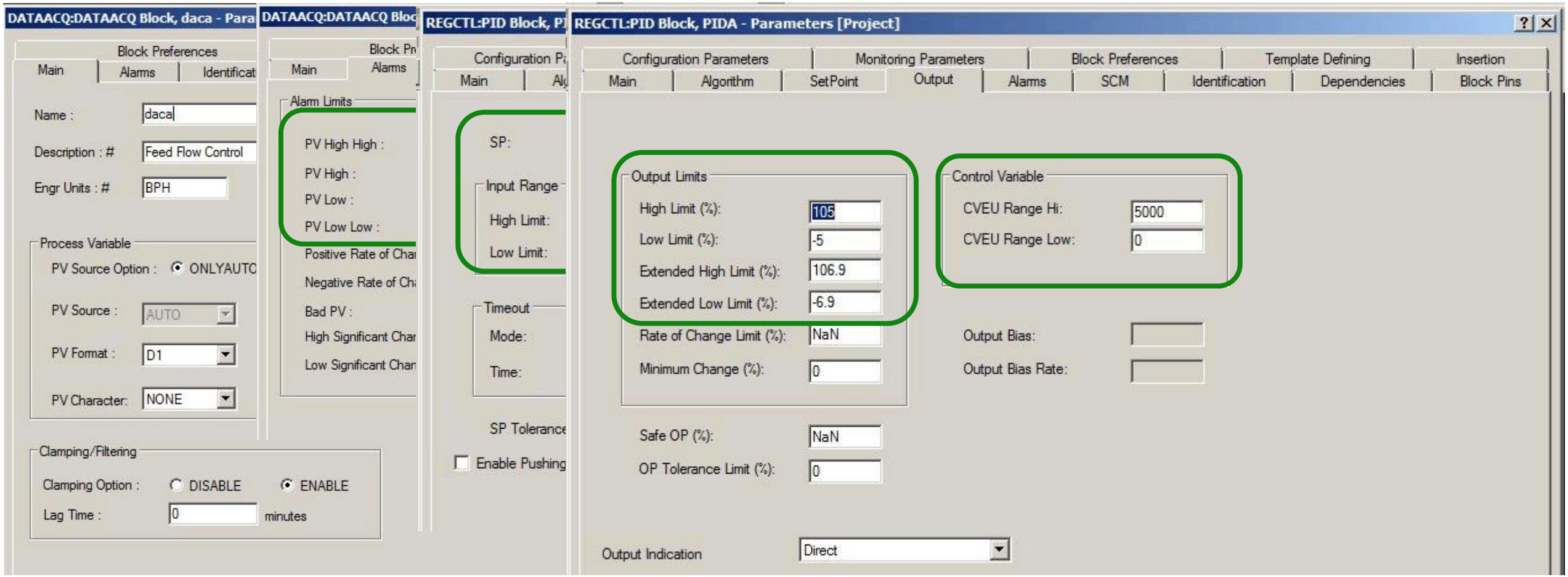
Point configuration is loaded into controller and stored on a DCS server

Point configuration



Point configuration is loaded into controller and stored on a DCS server

Point configuration



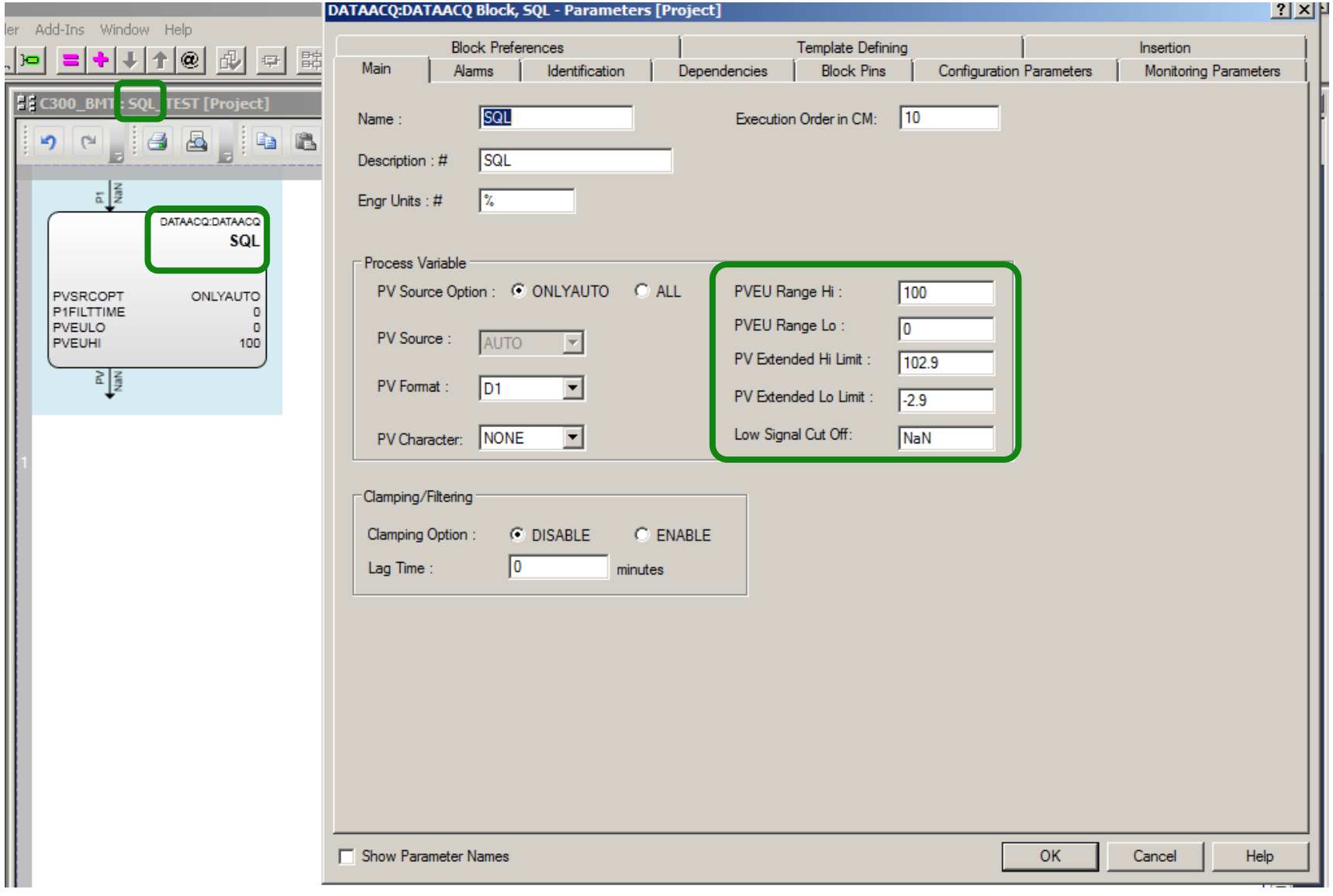
Point configuration is loaded into controller and stored on a DCS server

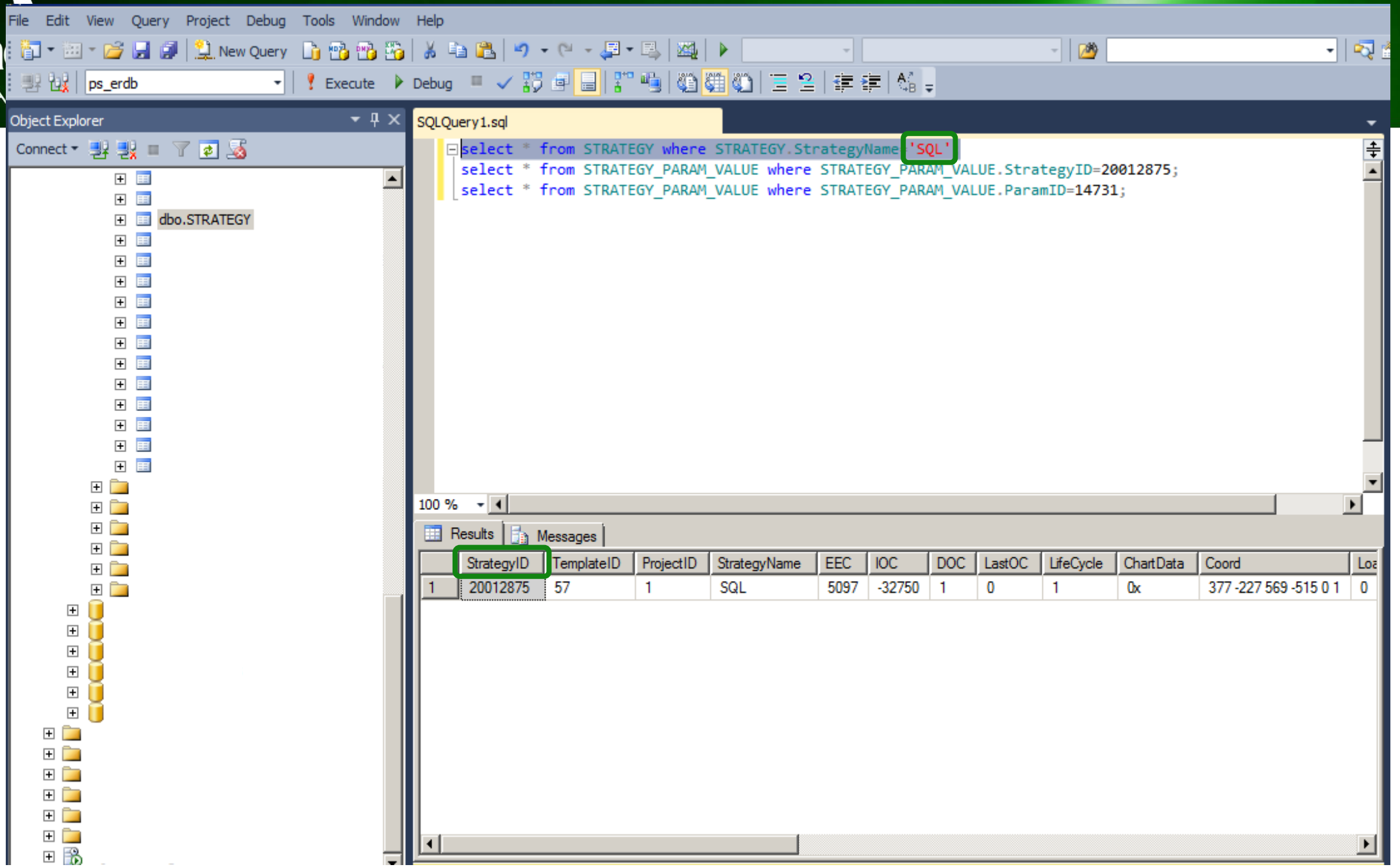
Retrieving point configuration

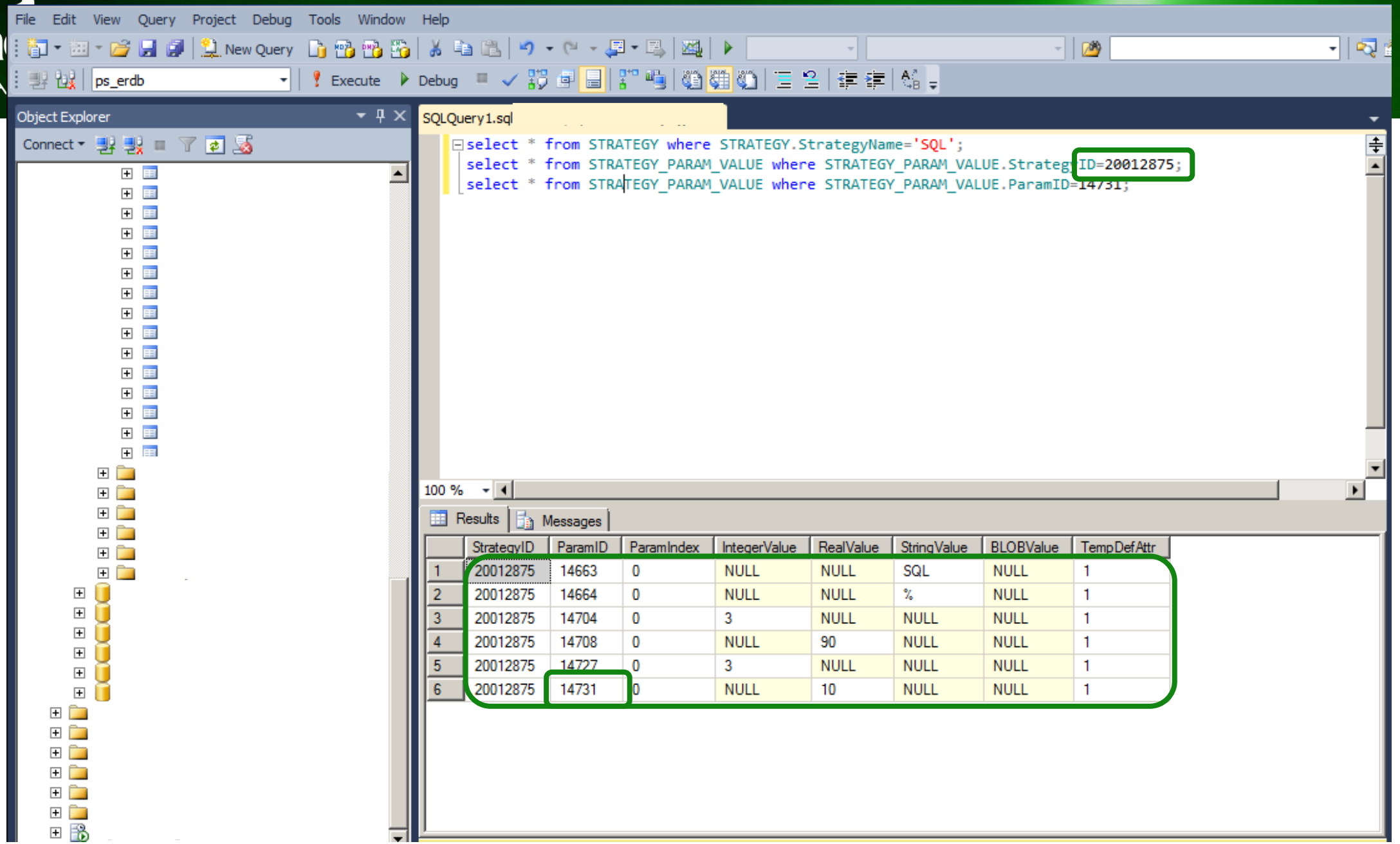
- **Directly from the controller**
  - DCS controllers are not easily obtainable to the attacker for analysis
- **Get access to engineering station and grab the project folder of interest**
  - Manual search, inconvenient
- **Query config from DCS Config DB**
  - Hundreds and hundreds of tables
  - Some DB entries may not have descriptions -> need to find the “manual”

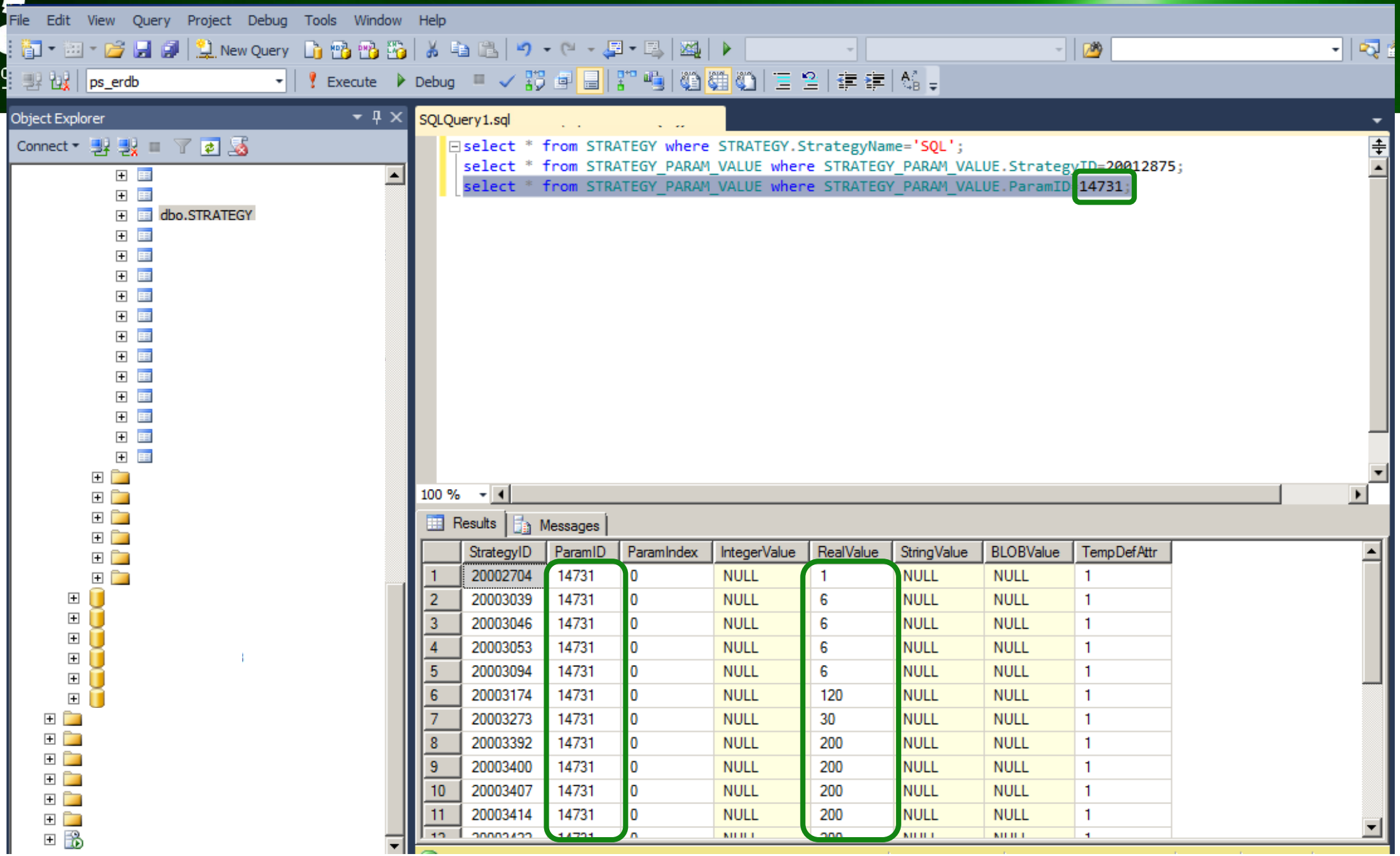
P.S. Honeywell’s manual on controller parameters is 2478 pages long. Happy reading!

Retrieving point configuration

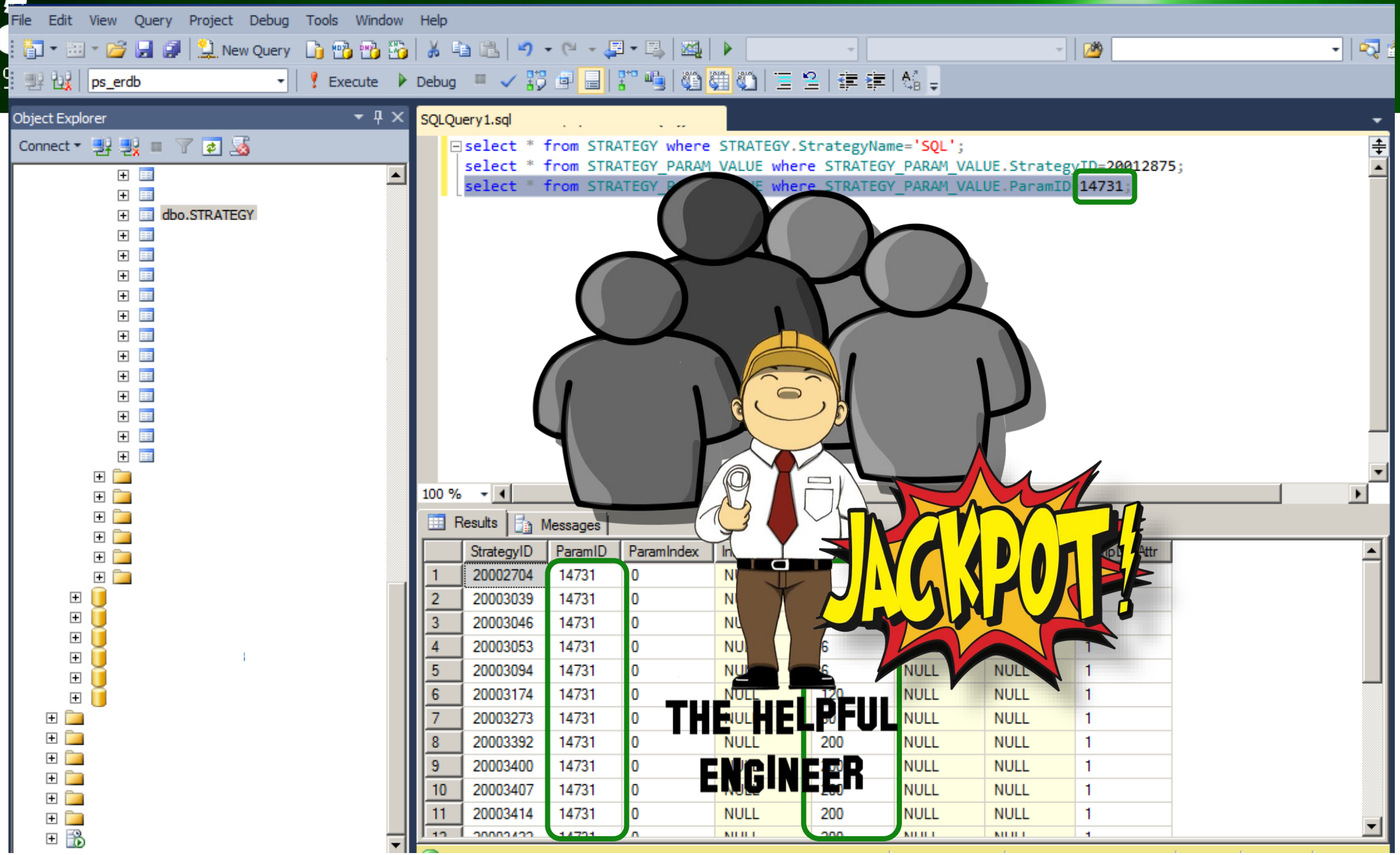



File Edit View Query Project Debug Tools Window Help
New Query
ps_erdb
Execute
Debug
Object Explorer
Connect
dbo.STRATEGY
SQLQuery1.sql
select * from STRATEGY where STRATEGY.StrategyName 'SQL'
select * from STRATEGY_PARAM_VALUE where STRATEGY_PARAM_VALUE.StrategyID=20012875;
select * from STRATEGY_PARAM_VALUE where STRATEGY_PARAM_VALUE.ParamID=14731;
100 %
Results
Messages
StrategyID TemplateID ProjectID StrategyName EEC IOC DOC LastOC LifeCycle ChartData Coord
1 20012875 57 1 SQL 5097 -32750 1 0 1 0x 377 -227 569 -515 0 1 0


File Edit View Query Project Debug Tools Window Help
New Query
ps_erdb
Execute
Debug
Object Explorer
Connect
SQLQuery1.sql
select * from STRATEGY where STRATEGY.StrategyName='SQL';
select * from STRATEGY_PARAM_VALUE where STRATEGY_PARAM_VALUE.StrategyID=20012875;
select * from STRATEGY_PARAM_VALUE where STRATEGY_PARAM_VALUE.ParamID=14731;
100 %
Results
Messages
StrategyID ParamID ParamIndex IntegerValue RealValue StringValue BLOBValue TempDefAttr
1 20012875 14663 0 NULL NULL SQL NULL 1
2 20012875 14664 0 NULL NULL % NULL 1
3 20012875 14704 0 3 NULL NULL NULL 1
4 20012875 14708 0 NULL 90 NULL NULL 1
5 20012875 14727 0 3 NULL NULL NULL 1
6 20012875 14731 0 NULL 10 NULL NULL 1



	StrategyID	ParamID	ParamIndex	IntegerValue	RealValue	StringValue	BLOBValue	TempDefAttr
1	20002704	14731	0	NULL	1	NULL	NULL	1
2	20003039	14731	0	NULL	6	NULL	NULL	1
3	20003046	14731	0	NULL	6	NULL	NULL	1
4	20003053	14731	0	NULL	6	NULL	NULL	1
5	20003094	14731	0	NULL	6	NULL	NULL	1
6	20003174	14731	0	NULL	120	NULL	NULL	1
7	20003273	14731	0	NULL	30	NULL	NULL	1
8	20003392	14731	0	NULL	200	NULL	NULL	1
9	20003400	14731	0	NULL	200	NULL	NULL	1
10	20003407	14731	0	NULL	200	NULL	NULL	1
11	20003414	14731	0	NULL	200	NULL	NULL	1



	StrategyID	ParamID	ParamIndex					
1	20002704	14731	0					
2	20003039	14731	0					
3	20003046	14731	0					
4	20003053	14731	0					
5	20003094	14731	0			NULL	NULL	1
6	20003174	14731	0			NULL	NULL	1
7	20003273	14731	0			NULL	NULL	1
8	20003392	14731	0	NULL	200	NULL	NULL	1
9	20003400	14731	0			NULL	NULL	1
10	20003407	14731	0			NULL	NULL	1
11	20003414	14731	0	NULL	200	NULL	NULL	1

Learning More from Use Cases

Miscellaneous : What's Different Plant by Plant

Diversity of architectures

- Plant is rarely operated with a help of a single DCS
  - Different plant units are operated by different DCSs, often of different vendors
  - Some units are operated by the PLC-based architectures
  - Old/legacy pieces of equipment
- Smaller plants or utilities are operated by non-homogeneous-vendor equipment configured by multiple integrators
- Specialized equipment or applications

Diversity of data scaling & formatting

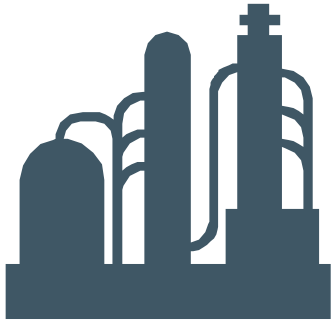
We interviewed more than 10 control engineers from multiple industries of different work experience globally

- Data scaling and formatting depends on multiple factors
  - Experience years of the control engineer
  - Equipment/application/protocol constraints
  - Requirements to data quality
  - Data normalization
  - Best practices (sometimes country/continent-dependent)
  - Customer preferences

Data loggers

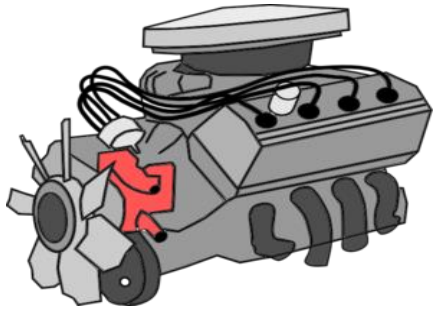
Configurations can be customized tailored to meet the scaling needs of a tremendous range of equipment and applications

Data normalization



PLANT 1

PLANT 2



Data normalization allows to compare data sets obtained in different scales or context

- Comparison of measurements from two distinct plants
- Communication of equipment working on different range and scale of measurements (e.g. different size/type of boilers in the plant)
- Monitoring performance of equipment
- Engineers perceive equipment performance faster/better when numbers are presented in % instead of actual EU

Anatomy of the Cyber-Physical Attack

From Script-Kiddie to Competent Attacker

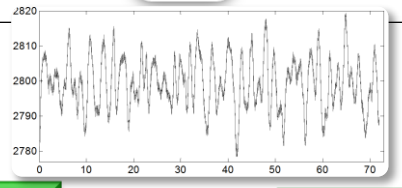
How to make this place going up in smoke?

Source: simentari.com

Cyber-physical attack

Cyber-Physical attack

1



1.1

Manipulate the process

- **Direct** → Set point change; manipulation of actuators
- **Indirect** → Deceiving controller/ operator about process state

1.2

Capture process feedback

- **Direct** → Direct observation of process values
- **Estimated or Derived** → From existing measurements or calculations

Most critical to success & hardest to achieve

2

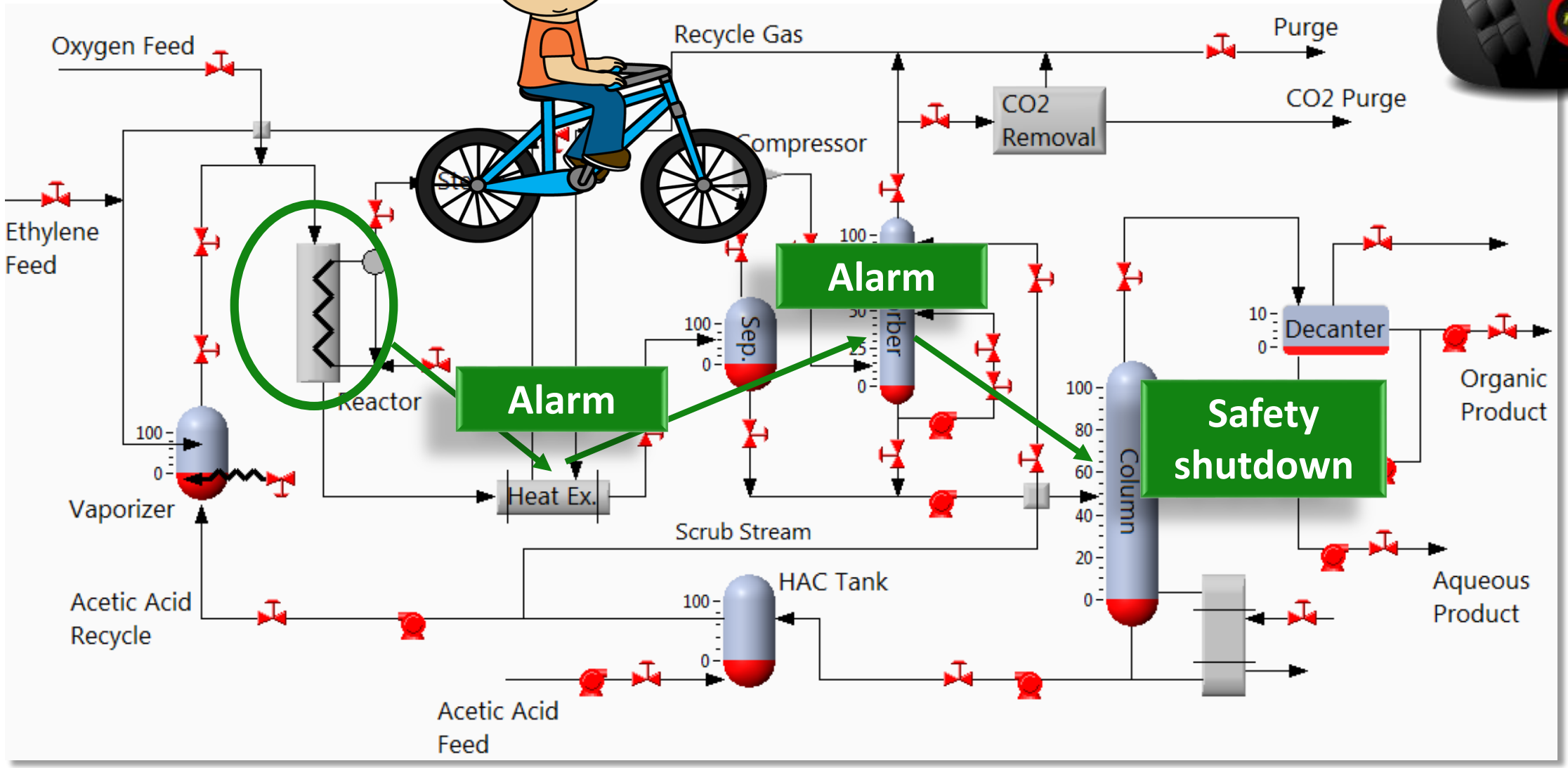
Not easy as well

Prevent response

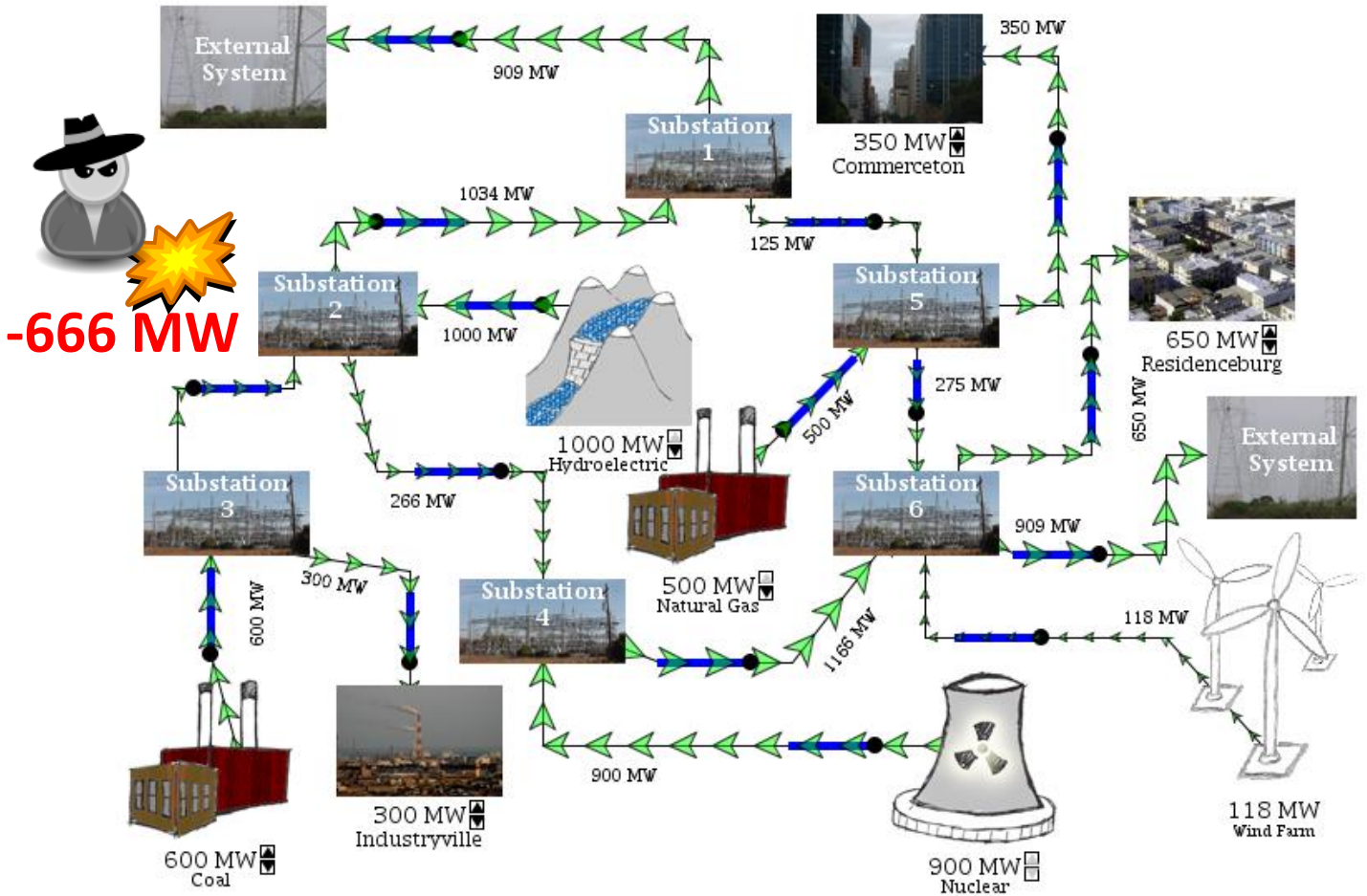
- **Operators** → Blind; Mislead

- **Control system** (including safety) → Modify operational/safety limits

Alarm propagation



State estimation in power sector



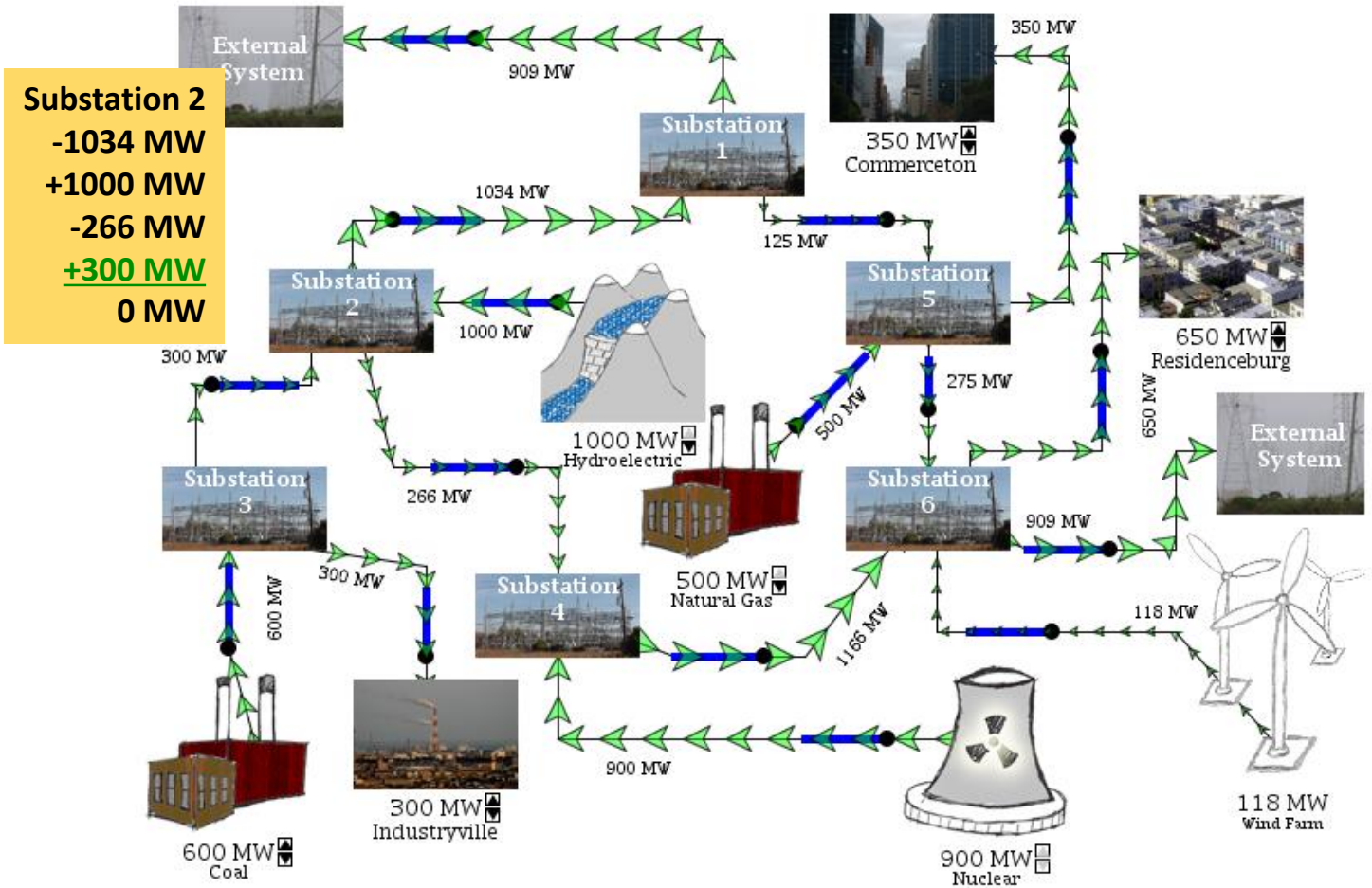
https://credc.mste.illinois.edu/applet/pg

- State Estimator (SE)
- Kirchoff's Current Law
  - Current flowing into a substation, group of substations, or a grid must equal current flowing out

P.S. Hire Ruben Santamarta to hack the SE

http://shinnai.altervista.org/papers_videos/STATG.pdf

State estimation in power sector



https://credc.mste.illinois.edu/applet/pg

- State Estimator (SE)
- Kirchoff's Current Law
  - Current flowing into a substation, group of substations, or a grid must equal current flowing out

P.S. Hire Ruben Santamarta to hack the SE

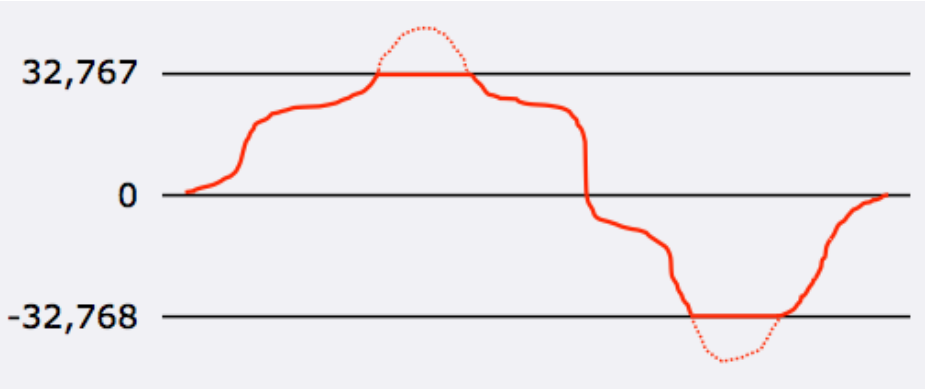
http://shinnai.altervista.org/papers_videos/STATG.pdf

Losing visibility into data

- The attacker pushes the process outside of normal operational envelope
  - She may lose visibility into process measurement
- Sensor calibration; signal clamping; truncation
- **Data scaling**
  - E.g. during process probing the attacker will make small changes to the process which may get “lost in translation”

5000089 -> scaled into 0-4095

5000089 -> floating point $5*10^6$



http://www.indiana.edu/~emusic/361/images/digitalaudio-clipping.png

Losing visibility into data

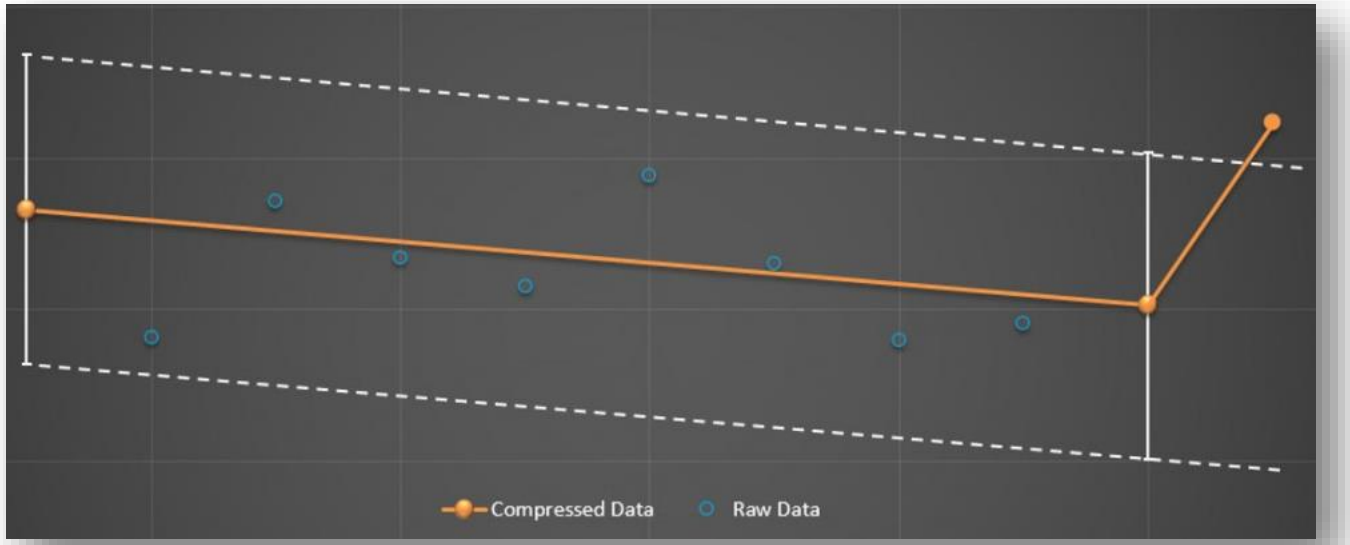
- The attacker pushes the process outside of normal operational envelope
  - She may lose visibility into process measurement
- Sensor calibration; signal clamping; truncation
- Data scaling
  - E.g. during process probing the attacker will make small changes to the process which may get “lost in translation”

5000089 -> scaled into 0-4095

5000089 -> floating point $5*10^6$

Where to monitor

- From the attacker standpoint single monitoring point is preferable
- By all means, the most hacker-friendly way to monitor process data in (RT)DB or Historian
- Historians typically rely on data compression for storage space optimization
  - "Unimportant" data is removed

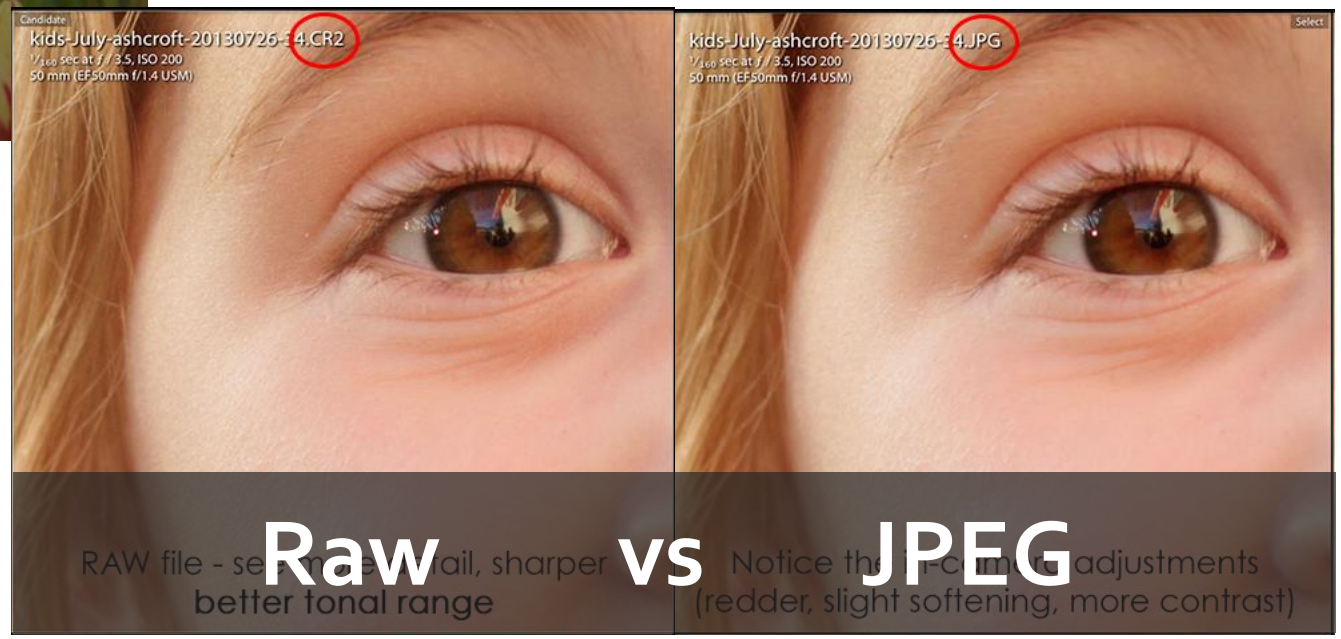


http://blog.dataparcsolutions.com/process-data-compression-why-its-a-bad-idea

Raw data vs. processed/translated data

http://www.the-amateur-photographer.com/raw-vs-jpeg/

IT DEPENDS



http://photographersconnection.com/should-you-photograph-in-raw-or-jpeg-lets-settle-this/

Where to monitor

- The problem with data compression is that data **LOST FOREVER**
  - Missing data is interpolated
- Historical data might not be appropriate for a feedback loop, especially for high precision attack
  - Because of lost data fidelity

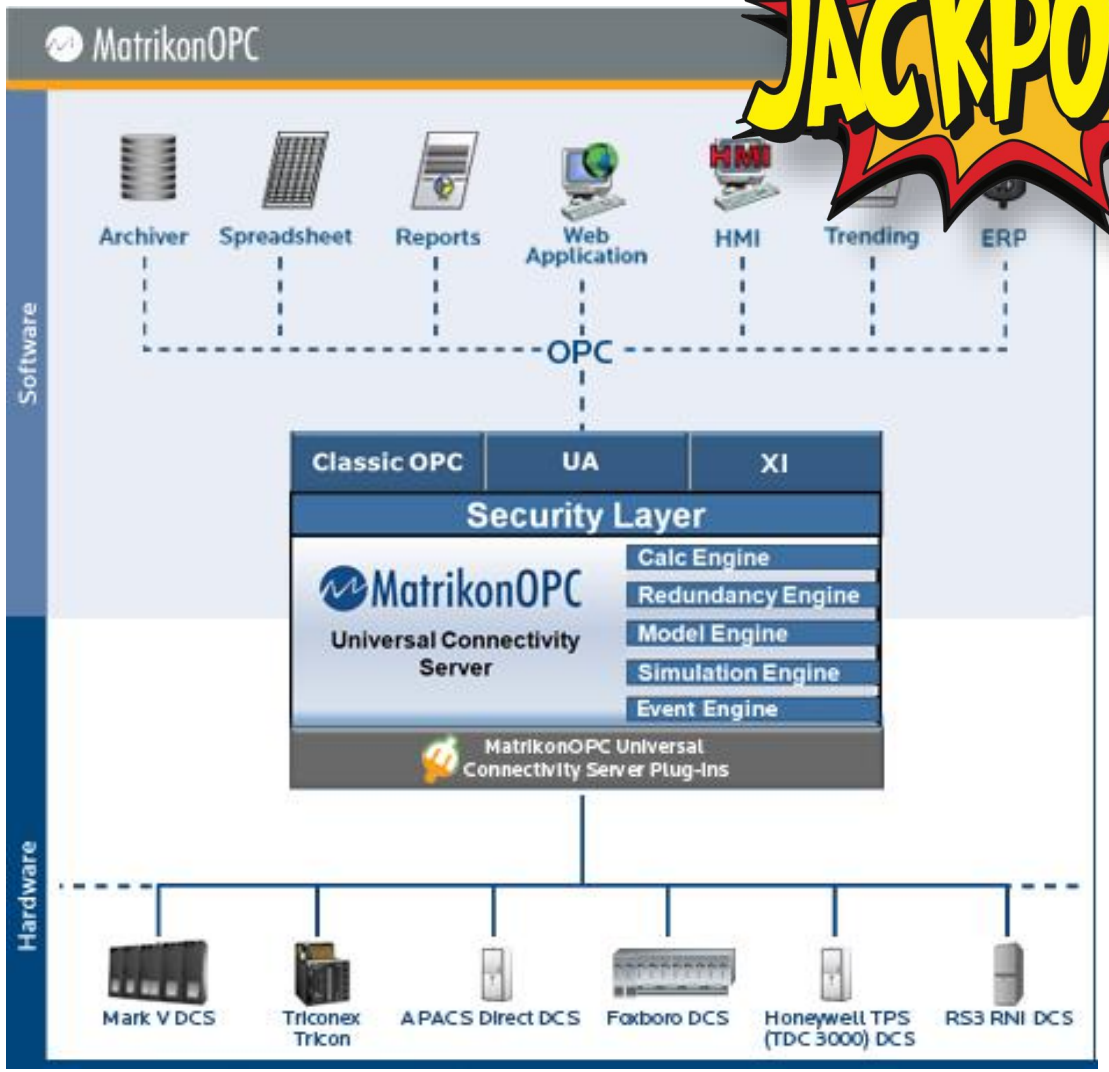
http://blog.dataparcsolutions.com/process-data-compression-why-its-a-bad-idea

Suppression of alarms

- **Alarms can be generated**
  - On the controller
  - In (some) DB
  - By a dedicated application
- May or may not be transmitted over wire

- Depending how the plant is configured, alarms can be suppressed in a dedicated application
  - Alarm shelving; changing priority of alarm; etc.
- **Cause alarm flood**
  - To be honest I have not idea (yet) how exactly to do it

OLE for Process Control (OPC)



HAVEX: Using OPC, the malware component gathers any details about OPC server and connected field devices and sends them back to the C&C.

- Query controllers for config data
- R/W configurable parameters
- Query process data; monitor alarms
- Issue control commands (if configured)
- In short, OPC allows achieving almighty privileges with minimal hacking efforts

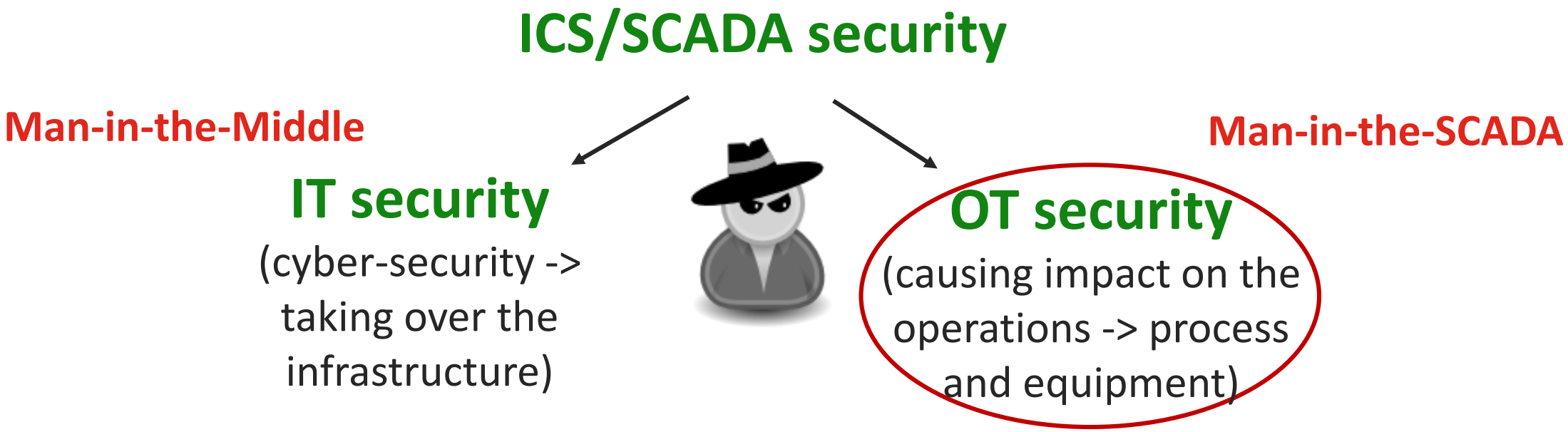
https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-176-02A

Key Takeaways

Turning this audience into ICS Superheros

Study the application under protection

- Once the access is gained to ICS infrastructure, the attack still needs to be performed
  - We need to do more applied research on understanding what the attacker needs to do and why



Key Take Aways

- **There are PERCEIVED and REAL threats in ICS world.** We need to challenge the assumptions about perceived threats

- Successful MITM attack requires a great deal of knowledge about data point configuration
  - It involves extensive reconnaissance and specialized knowledge

- **Everything what is marked as** 

must be protected more conservatively than the prisoners in high-security correctional facilities
  - Lock away config files, monitor access
  - Harden DCS/SCADA servers
  - Upgrade OPC to OPC UA (please)

Goal: New line of thinking

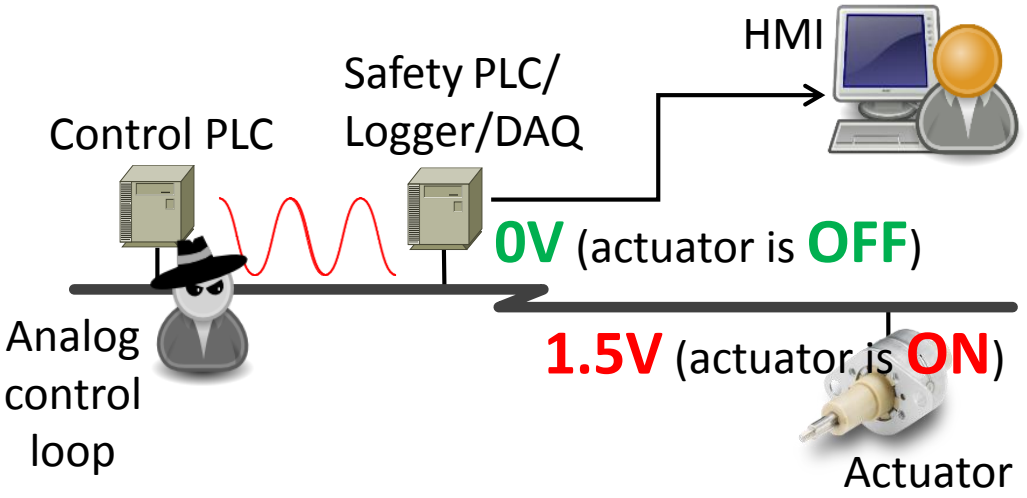
- **Understanding point configuration fundamentals reveals an additional attack surface**
- Instead of modifying data directly

1. **Modify the configuration of the data point**
   - Change sensor calibration or its range. Good for alarm suppression and blinding operators & controllers
2. **Take advantage of it**

Taking advantage of point config

Never Trust Your Inputs: Causing 'Catastrophic Physical Consequences' from the Sensor (or how to fool ADC)

A. Bolshev & M. Krotofil. Black Hat Asia 2016



Any Q?

Marina Krotofil
marina.krotofil@honeywell.com
@marmusha

Honeywell

Chris Sistrunk
chris.sistrunk@mandiant.com
@chrissistrunk

MANDIANT®
A FireEye® Company