Automated testing of protection with Omicron and Python

07-02-2024

STEDIN.NET | VOOR DE NIEUWE ENERGIEGENERATIE

ABOUT STEDIN

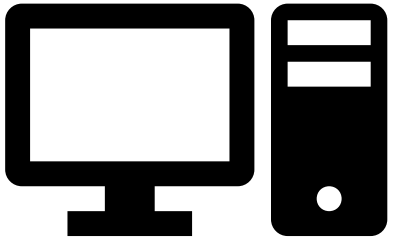
- Distribution grid operator in most of South Holland and the provinces Utrecht and Zeeland
- More than 2.3 million households
- Our grid in numbers :

Type	Amount
Primary (25/60/66 kV)	~186
MV Transport (10/13/21/23 kV)	~400
Secondary (0.4 kV)	~23,000

Testing of IEDs

- IEDs are the backbone of the substation
- Proper operation of the protection functions is of utmost importance
- Each IED is tested for correct operation of every protection function that is active:
  - Thresholds
  - Reaction times
- Testing is mainly done “inside” and with secondary currents / voltages
  - Primary values are used only in special occasions
- Omicron hardware and software are used as testing equipement

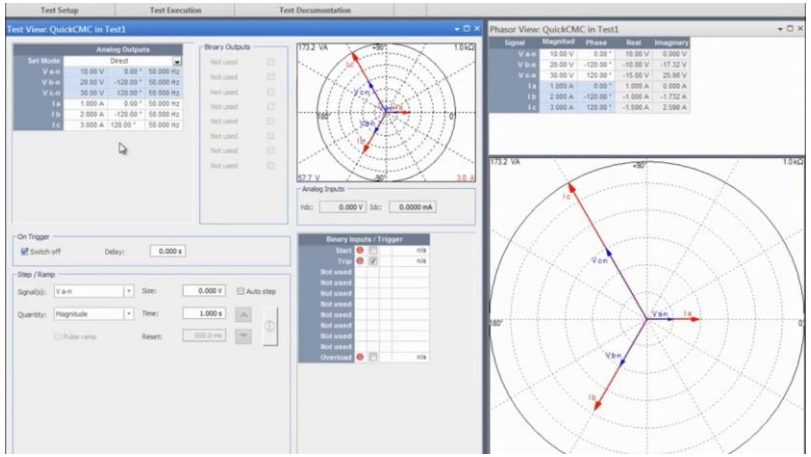
Testing setup



U

I

OCC file



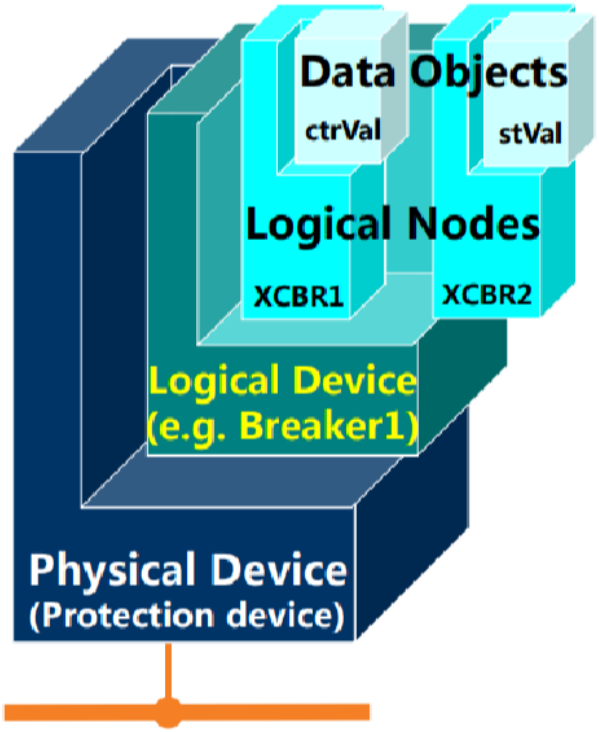
Verifying IED operation and reaction times

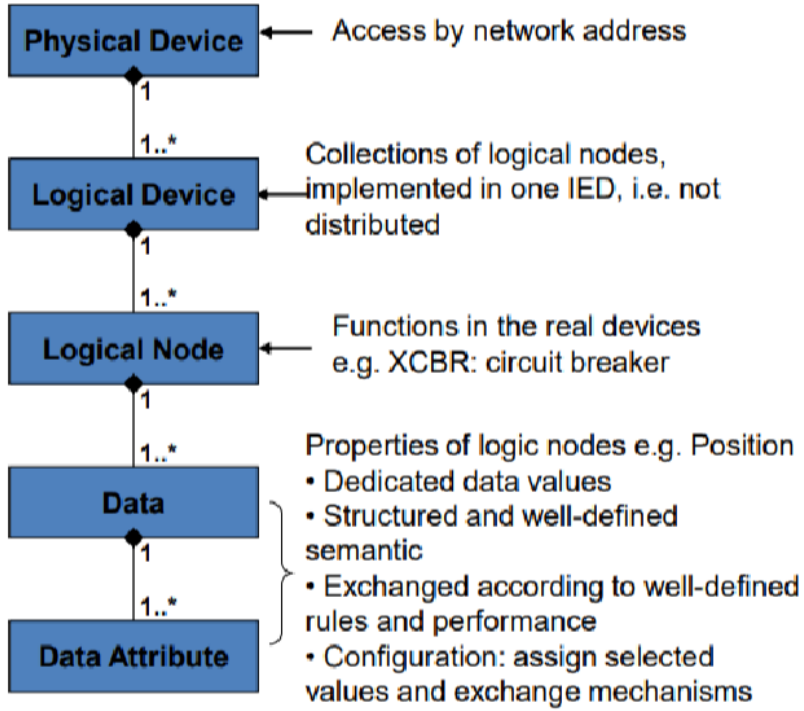
IEC 61850
IN A NUTSHELL (1)

DATA MODEL

Unified names for (amongst others):

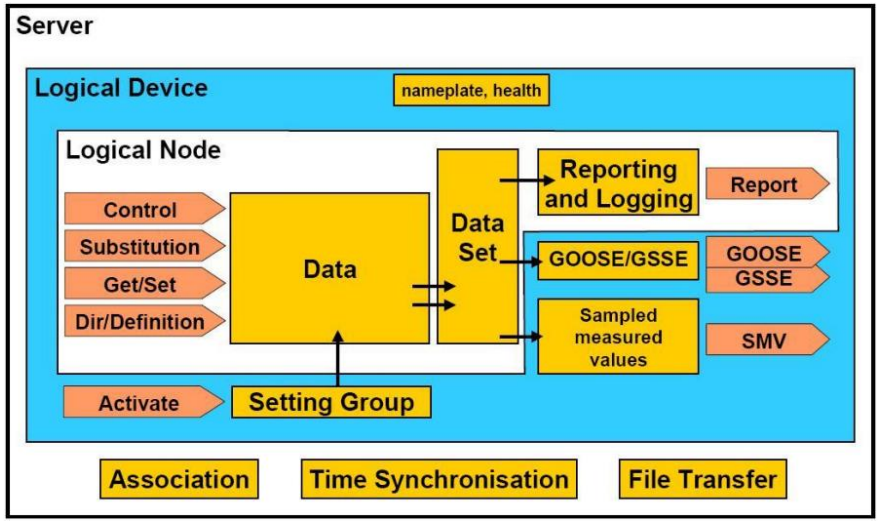
- PD – Voltage level/section/IED
- LD – Object group (CTRL/PROT)
- LN - Object/function (XCBR/PTOC)
- DO – Details (stVal/Str.phsA)





DT1K1Q02A1/PTOC.Str.phsA (104 = 666958)

IEC 61850
IN A NUTSHELL (2)



COMMUNICATION

MMS (Manufacturing Message Specification)

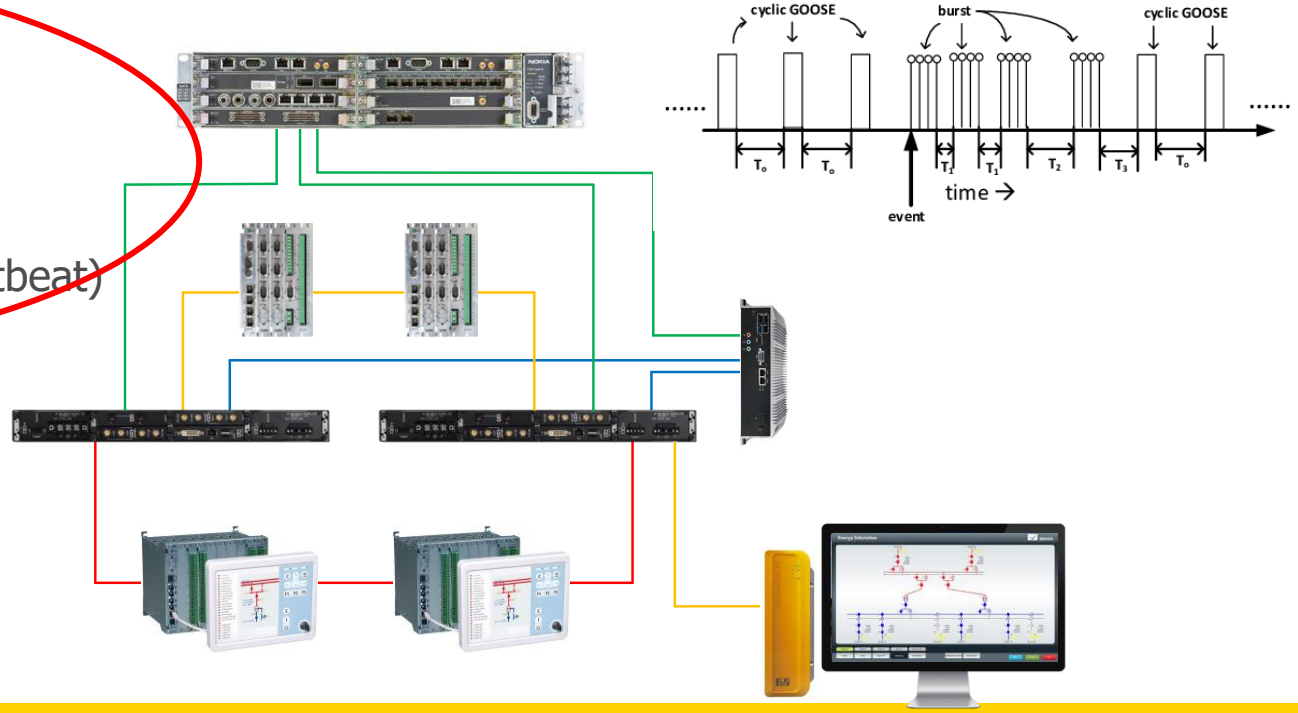
- Based on TCP/IP – routable across networks
- Client/server – communication is verified

GOOSE (Generic Object Oriented Substation Event)

- Based on Ethernet – local network
- Multicasts – messages are published
- Direct sending for events and periodic messages (heartbeat)

SV (Sampled Values)

- Based on Ethernet – local network
- Multicasts – one-way communication
- Flow of messages with sampled ratings of 4000/12800 samples/sec – 5,3/12,5 Mb/sec



Omicron
Hard-wired vs. GOOSE

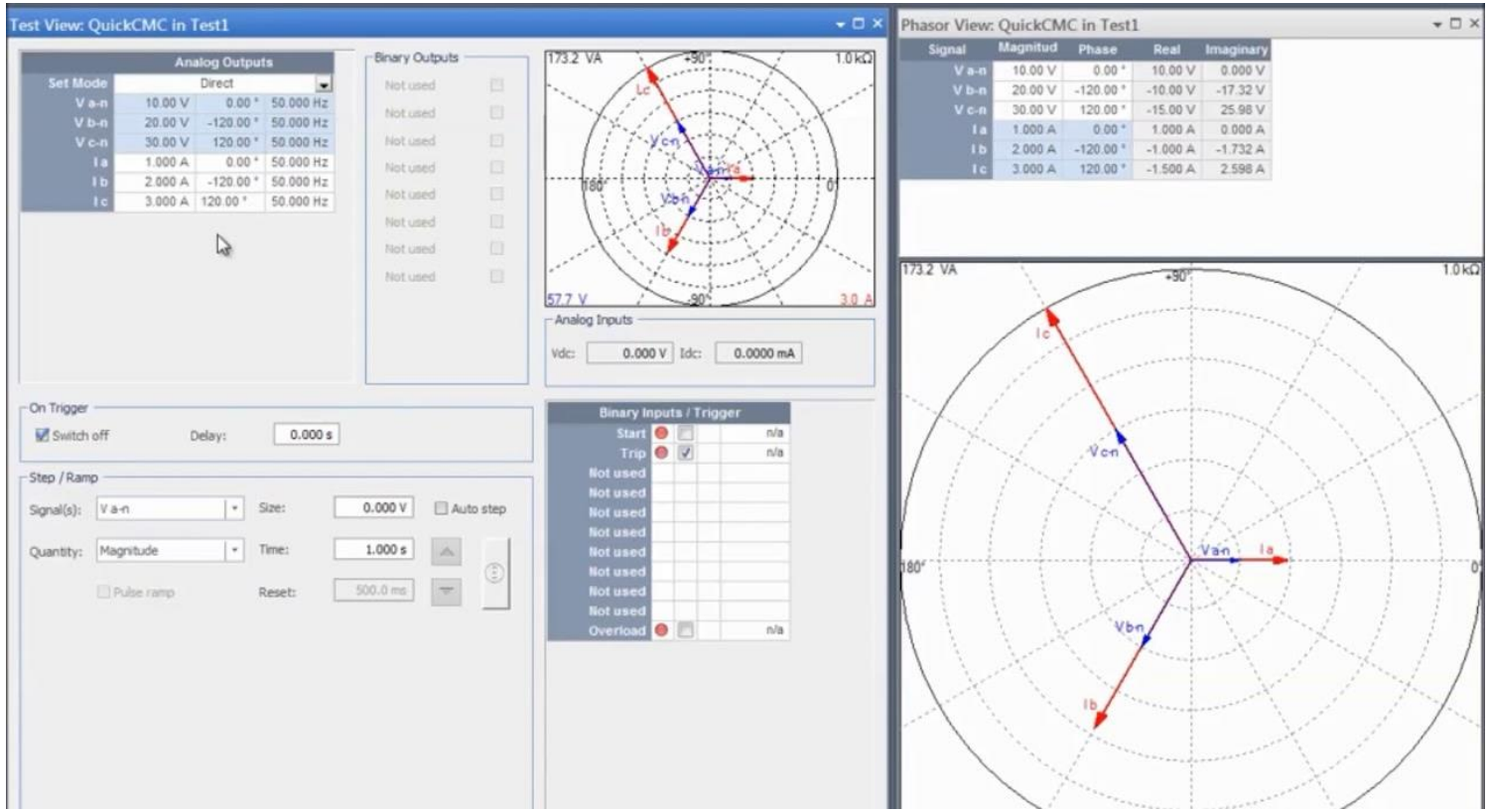
PROTECTION, TIME MEASURING AND TRIPS

Omicron
HARD-WIRED VS. GOOSE

PROTECTION TESTING WITH HARD-WIRED CONNECTIONS

Measuring the reaction time of the start/trip contacts using hard-wired connections

- All protection functions send start/trip signals
- Often only testing per protection functions, disabling conflicting protection functions
- **<u>Testing distributed protection schemes is complex and time-consuming</u>**

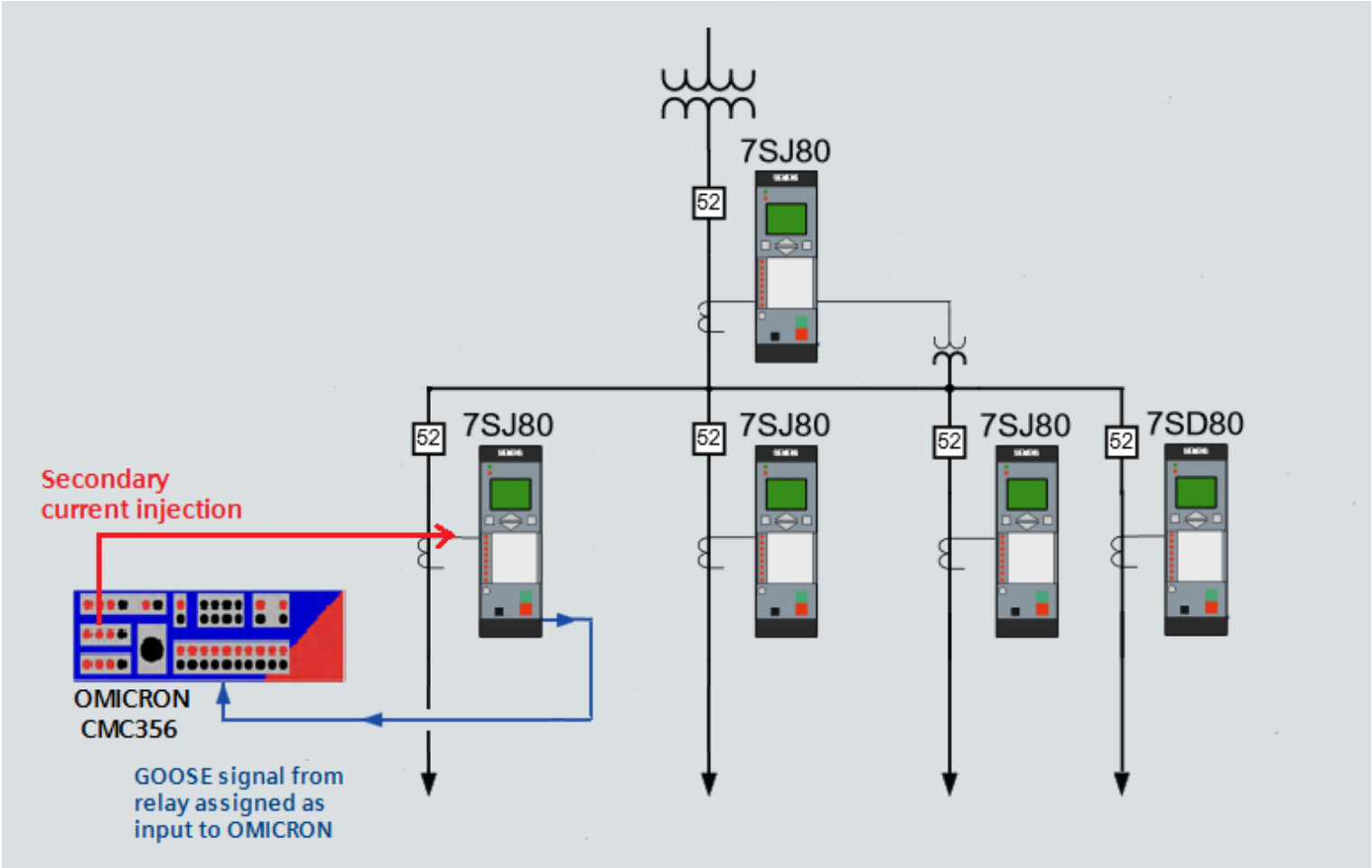


Omicron
HARD-WIRED VS. GOOSE

PROTECTION TESTING WITH GOOSE

Measuring the reaction times of the start/trip signals with GOOSE

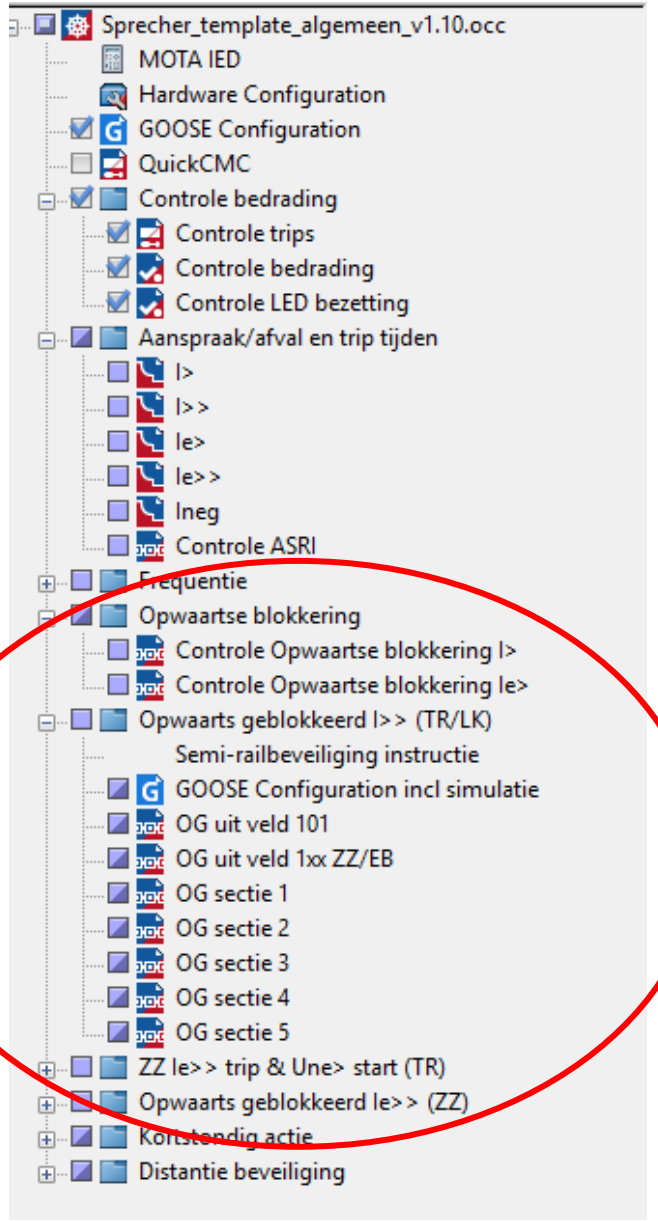
- Time measurements per protection function (for example I> or I>>).
- Subscriptions, as well as simulations of GOOSE messages, are possible
- Testing of distributed protection schemes is very easy by sending the signals over the network
  (for example reverse-blocking protection)



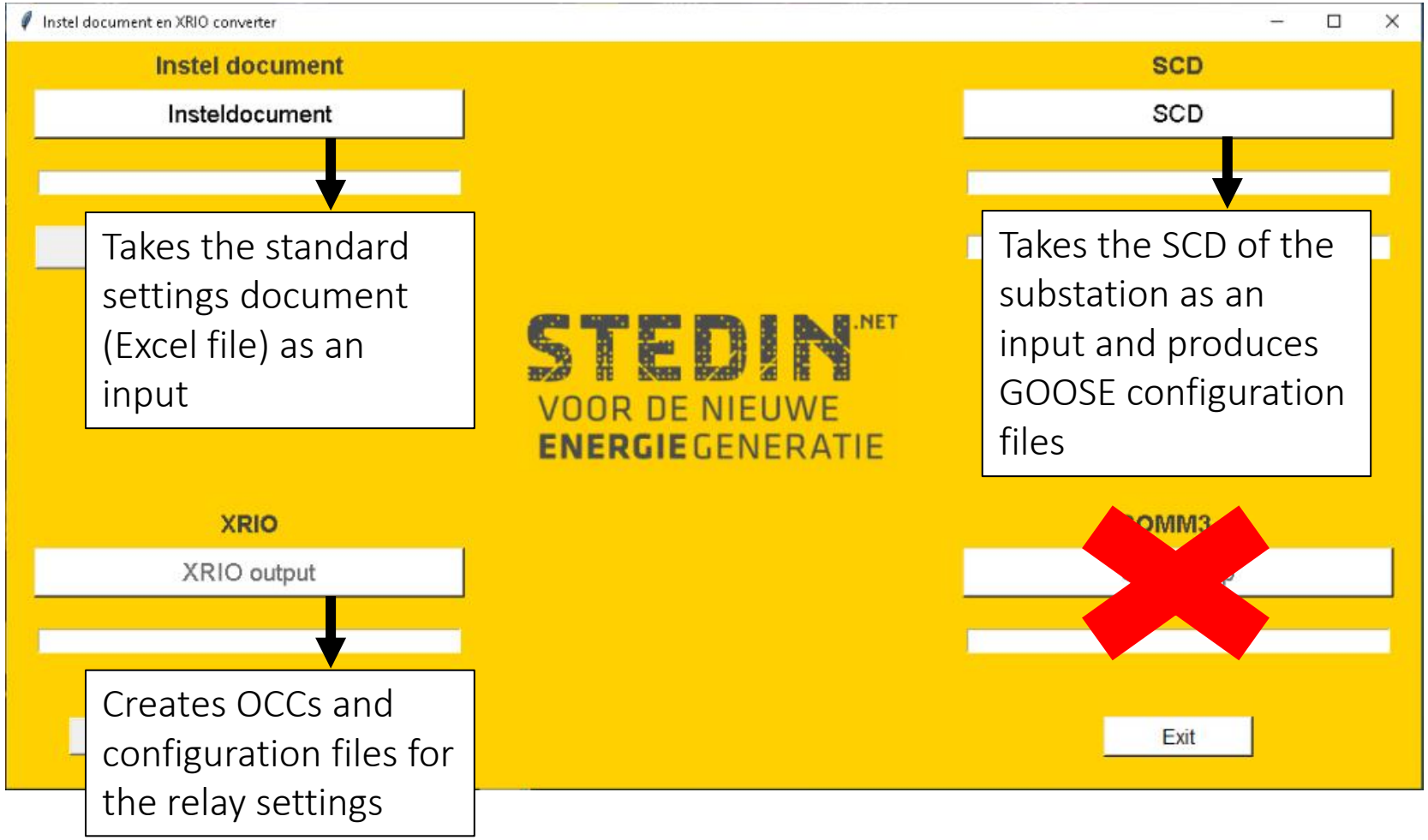
HOW WE AUTOMATE IN STEDIN

Standard OCC template

- Adjusted to the standard protection functions that we use
- GOOSE configuration included
- Distributed protection schemes also present



Tooling - NA beveiligingApp



Creating XRIO files

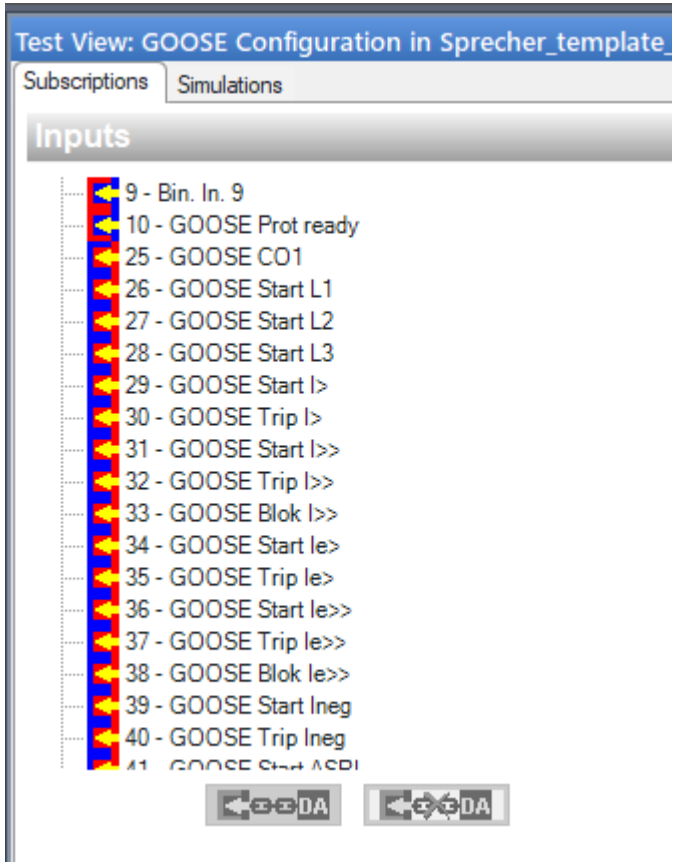
Spannings-	Veldnummer		Veldtype	Richting	I > Pri	I > Sec	t >	I >> Pri	I >> Sec	t >>	Ie> Pri	Ie> Sec	t Ie>	Bepaling Ie	Ie>> Pri	Ie>> Sec	t Ie>>	Ineg Pri	Ineg Sec	t Ineg	Opw I>>	Opw Ie>>	Bepaling ASRI	Frequentie trip	Bepaling ASRI
niveau	Oud	Nieuw			[A]	[A]	[s]	[A]	[A]	[s]	[A]	[A]	[s]		[A]	[A]	[s]	[A]	[A]	[s]	Blokk	Blokk		[Hz]	
10kV	-	101	Afgaand veld	Reserve	90	1.13	0.30	120	1.50	0.00	60	0.75	0.00								OB I>	OB Ie>		48.4	
10kV	-	102	Afgaand veld	Groenhovenweg Zuid	360	4.50	0.60	2500	31.25	0.30	100	1.25	0.60								OB I>	OB Ie>		48.4	
10kV	-	103	Afgaand veld	Reserve	90	1.13	0.30	120	1.50	0.00	60	0.75	0.00								OB I>	OB Ie>		48.4	
10kV	-	104	Afgaand veld	Nieuwe Gouwe O.Z. 18	360	4.50	0.60	2500	31.25	0.30	100	1.25	0.60								OB I>	OB Ie>		48.4	

Invul blok

Stat	Name	ID	Foreign ID	Description	Reference value	Value	Unit	Min	Max	Formula
✓	USER_NAME	USER_NAME		Gebruikersnaam	Nicola Tesla					
✓	STATION	STATION		Stationsnaam	Stationnaam	Test				
✓	STATIONNR	STATIONNR		Stationsnummer	TX1234	TX12				
✓	UNOM	UNOM		Spanningsniveau (pri)	10.5	50.0	kV	0.0	+inf	
✓	UNOM_SEC	UNOM_SEC		Nominale spanning (sec)	110	100	V	0	250	
✓	GEAARD	GEAARD		Geaard net	☑	☑				
✓	VELDNR	VELDNR		Veldnummer	101	102				
✓	VELDTYPE	VELDTYPE		Type veld	Afgaand veld	Afgaand veld				
✓	RICHTING	RICHTING		Richtingnaam	Richting	Richting				
✓	TYPE_BEVEILIGING	TYPE_BEVEILIGING		Type beveiliging	OMT	OMT				
✓	LEVERANCIER	LEVERANCIER		Leverancier	Sprecher	Sprecher				
✓	FIRMWARE	FIRMWARE		Firmwareversie IED	8.64k					
✓	SERIENR	SERIENR		Serienummer van IED	33110525/G002					
✓	MOTA_TYPE	MOTA_TYPE		MOTA Type	Type C1	Type C1				
✓	IP	IP		IP addres	10.0.0.0					
✓	I_PRI	I_PRI		Primaire stroom van de CT	600	1200	A	0	+inf	
✓	I_SEC	I_SEC		Secundaire stroom van de CT	1	1				
✓	CT_SP_BUS	CT_SP_BUS		CT sterpunt richting rail (is true)	☐	☑				
✓	SETTINGGROUP	SETTINGGROUP		Settinggroup	1	1		0	4	
✓	I1_PRI	I1_PRI		I> primair	300	0	A	0	+inf	
✓	I1_T	I1_T		I> tijd	0.9	0.0	s	0.0	+inf	
✓	I2_PRI	I2_PRI		I>> primaire	1200	0	A	0	+inf	
✓	I2_T	I2_T		I>> tijd	0.0	0.0	s	0.0	+inf	
✓	Ie_PRI1	IE_PRI1		Ie> primair	150	0	A	0	+inf	
✓	IE_T	IE_T		Ie> tijd	0.5	0.0	s	0.0	+inf	

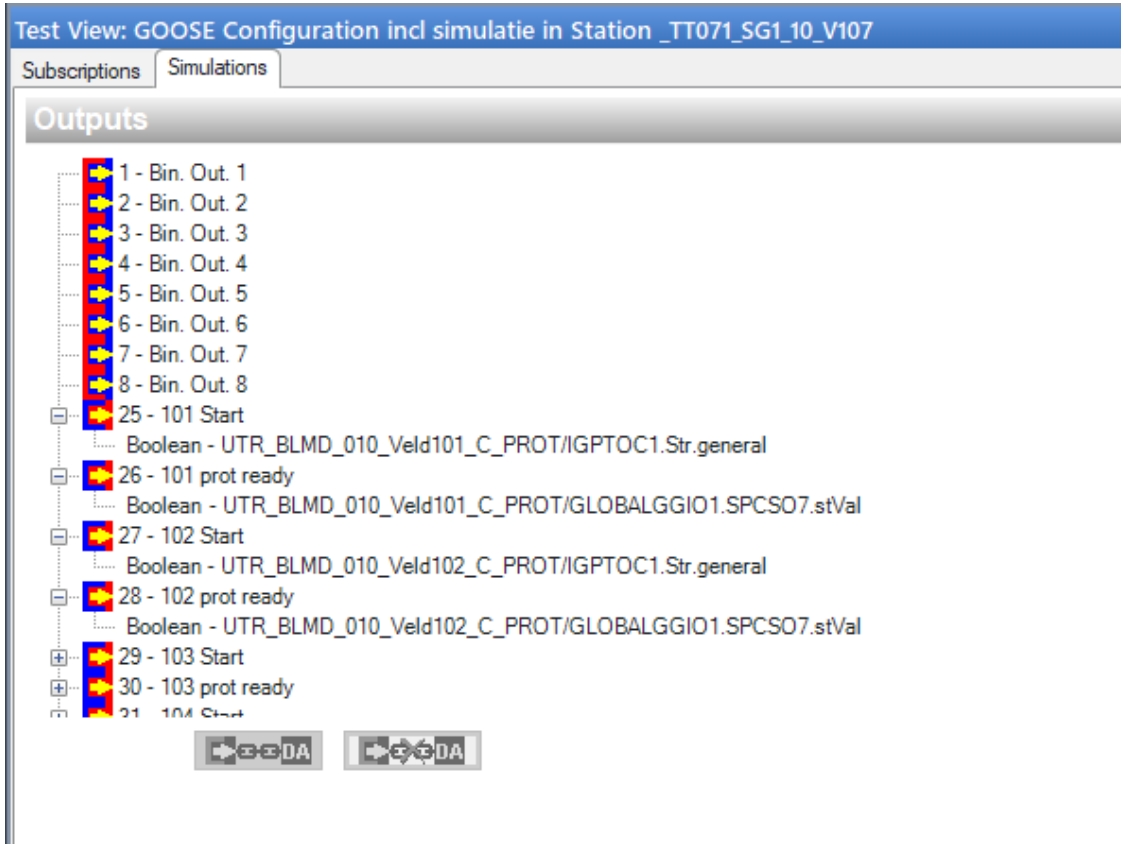
Creating GOOSE configuration files (1)

```
<?xml version="1.0" encoding="UTF-8"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" revision="B" version="2007">
    <Header id="" revision="1" toolID="SPRECON-E IEC61850 Mapper 2.34 SP1 (3099)" version="1">
        <History>
            <Hitem revision="1" version="1" what="Initial creation of model." when="2022-12-02 15:15:10" who="SPRECON-E IEC61850 Mapper 2.34 SP1 (3099) user::Veikko.Pluijm resources::com.sprecher.easyeng.resources-7.0.10.v2021112
        </History>
    </Header>
    <Substation name="NLSTEDUTRECHBLMD">
        <VoltageLevel name="VOLTAGELEVEL">
            <Bay name="BAY"/>
        </VoltageLevel>
    </Substation>
    <Communication>
        <SubNetwork name="Sprecon-Network" type="8-MMS">
            <BitRate multiplier="M" unit="b/s">100</BitRate>
            <ConnectedAP apName="AP1" iedName="UTR_BLMD_010_GW_X_">
                <Address>
                    <P type="IP">10.66.225.11</P>
                    <P type="IP-SUBNET">255.255.255.0</P>
                    <P type="IP-GATEWAY">10.66.225.1</P>
                </Address>
                <GSE cbName="gob_C" ldInst="CTRL">
                    <Address>
                        <P type="MAC-Address">01-0C-CD-01-00-01</P>
                        <P type="APPID">0002</P>
                        <P type="VLAN-ID">00A</P>
                        <P type="VLAN-PRIORITY">4</P>
                    </Address>
                    <MinTime multiplier="m" unit="s">50</MinTime>
                    <MaxTime multiplier="m" unit="s">1000</MaxTime>
                </GSE>
                <GSE cbName="gob_D" ldInst="GENERAL">
                    <Address>
                        <P type="MAC-Address">01-0C-CD-01-00-01</P>
                        <P type="APPID">0003</P>
                        <P type="VLAN-ID">00A</P>
                        <P type="VLAN-PRIORITY">4</P>
                    </Address>
                    <MinTime multiplier="m" unit="s">50</MinTime>
                    <MaxTime multiplier="m" unit="s">1000</MaxTime>
                </GSE>
            </ConnectedAP>
            <ConnectedAP apName="AP1" iedName="UTR_BLMD_010_SC_X_">
                <Address>
                    <P type="IP">10.66.225.12</P>
                    <P type="IP-SUBNET">255.255.255.0</P>
                    <P type="IP-GATEWAY">10.66.225.1</P>
                </Address>
                <GSE cbName="gob_N" ldInst="GENERAL">
                    <Address>
                        <P type="MAC-Address">01-0C-CD-01-00-01</P>
                        <P type="APPID">0008</P>
```

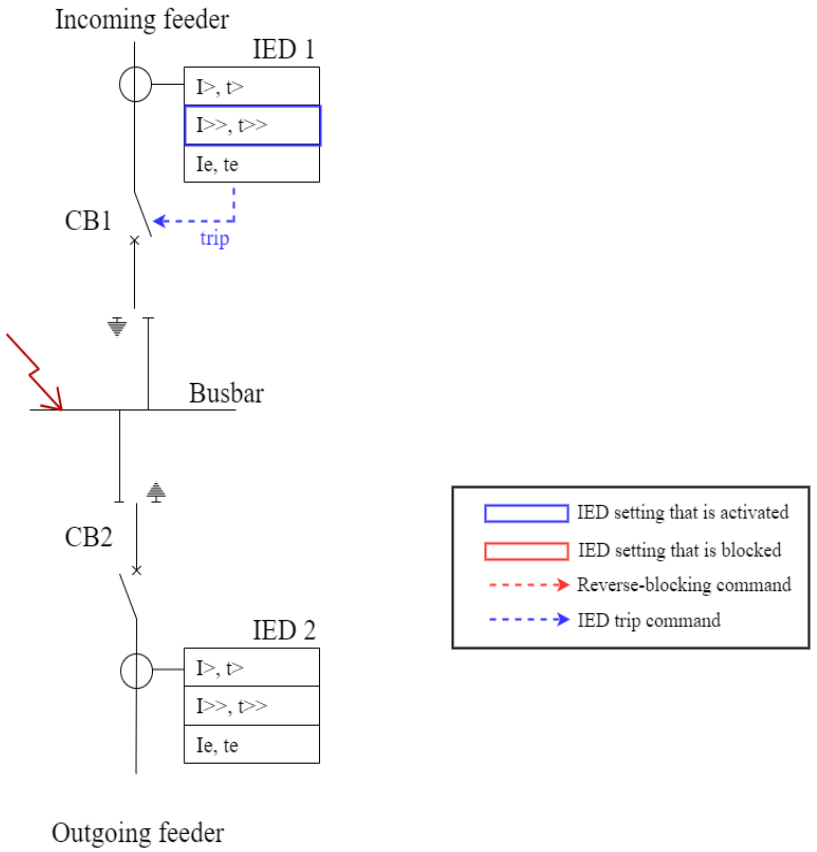
Creating GOOSE configuration files (2)

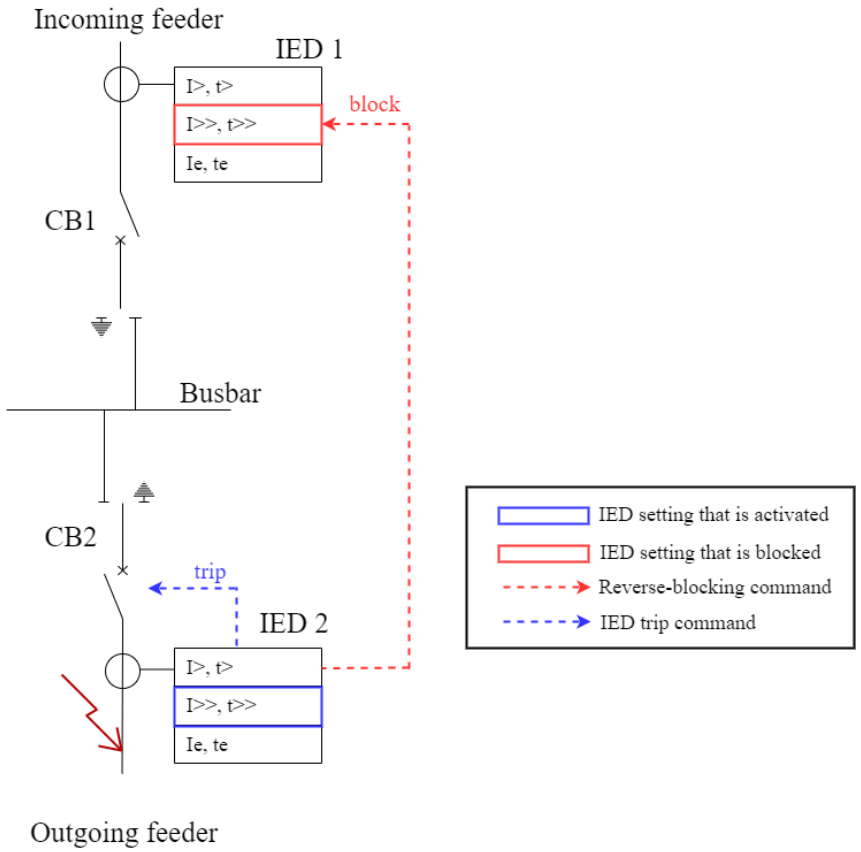
BUT ALSO….SIMULATION OF GOOSE



Example CASE - REVERSE-BLOCKING Scheme (1)

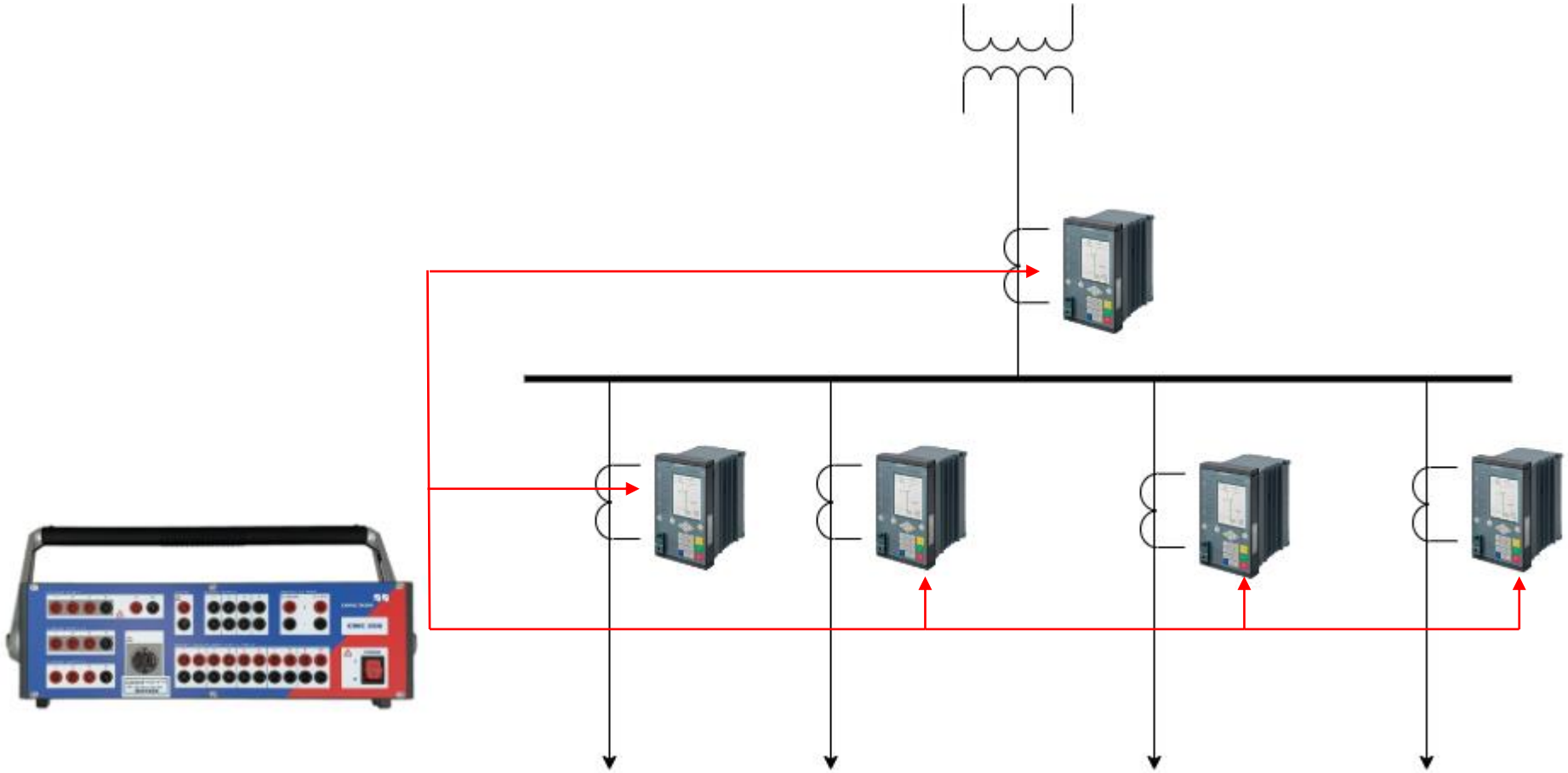
THEORY





Example CASE - REVERSE-BLOCKING Scheme (2)

HARD-WIRED TESTING



Example CASE - REVERSE-BLOCKING Scheme (3)

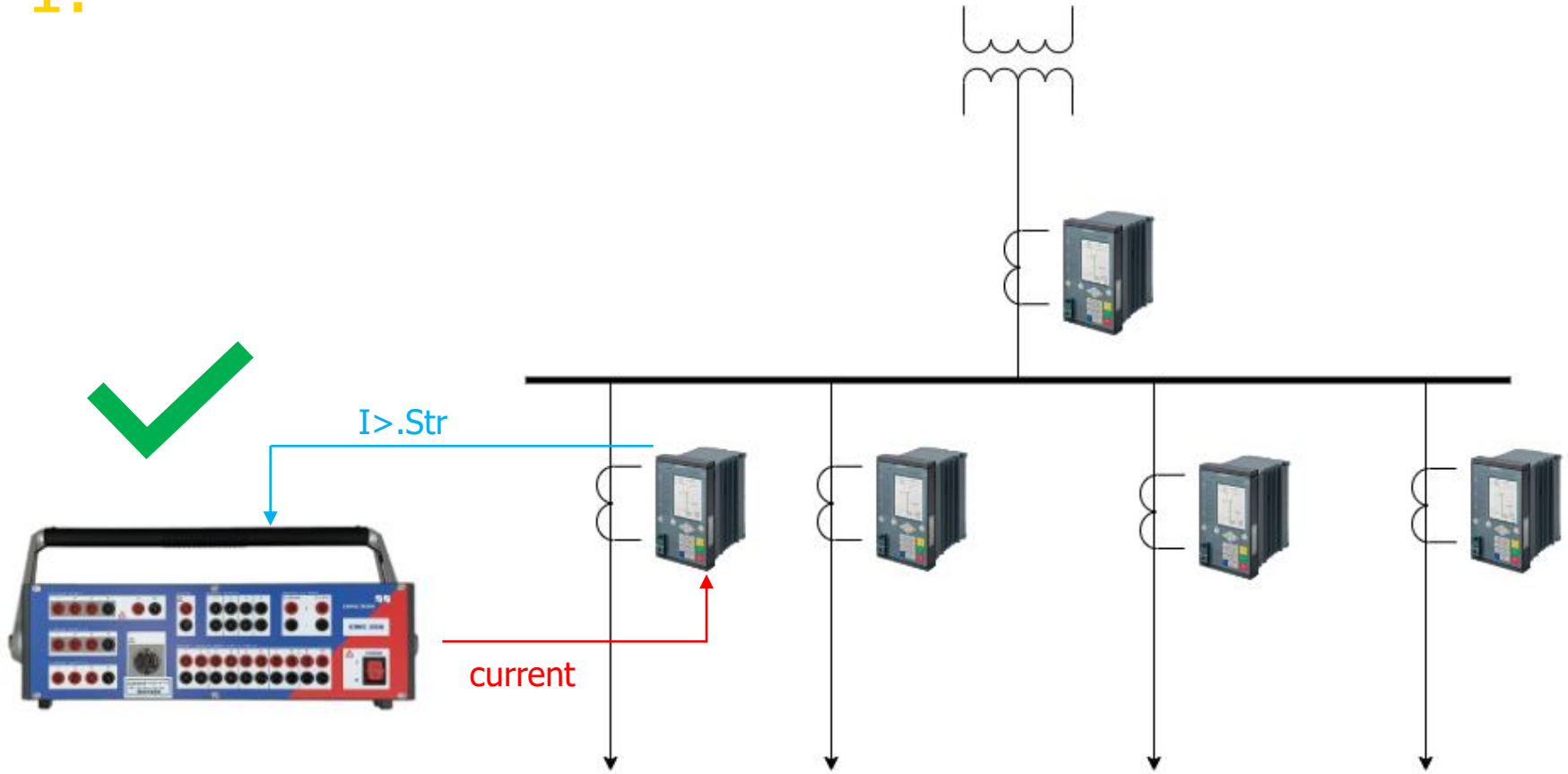
GOOSE-BASED TESTING

- Divided in two parts
- Part 1: Whether the outgoing feeders send the correct GOOSE messages for blocking
- Part 2: Whether the incoming feeders are actually blocked at the receiving of those GOOSE messages
- Faster testing in this manner due to less rewiring

Example CASE - REVERSE-BLOCKING Scheme (3)

GOOSE-BASED TESTING

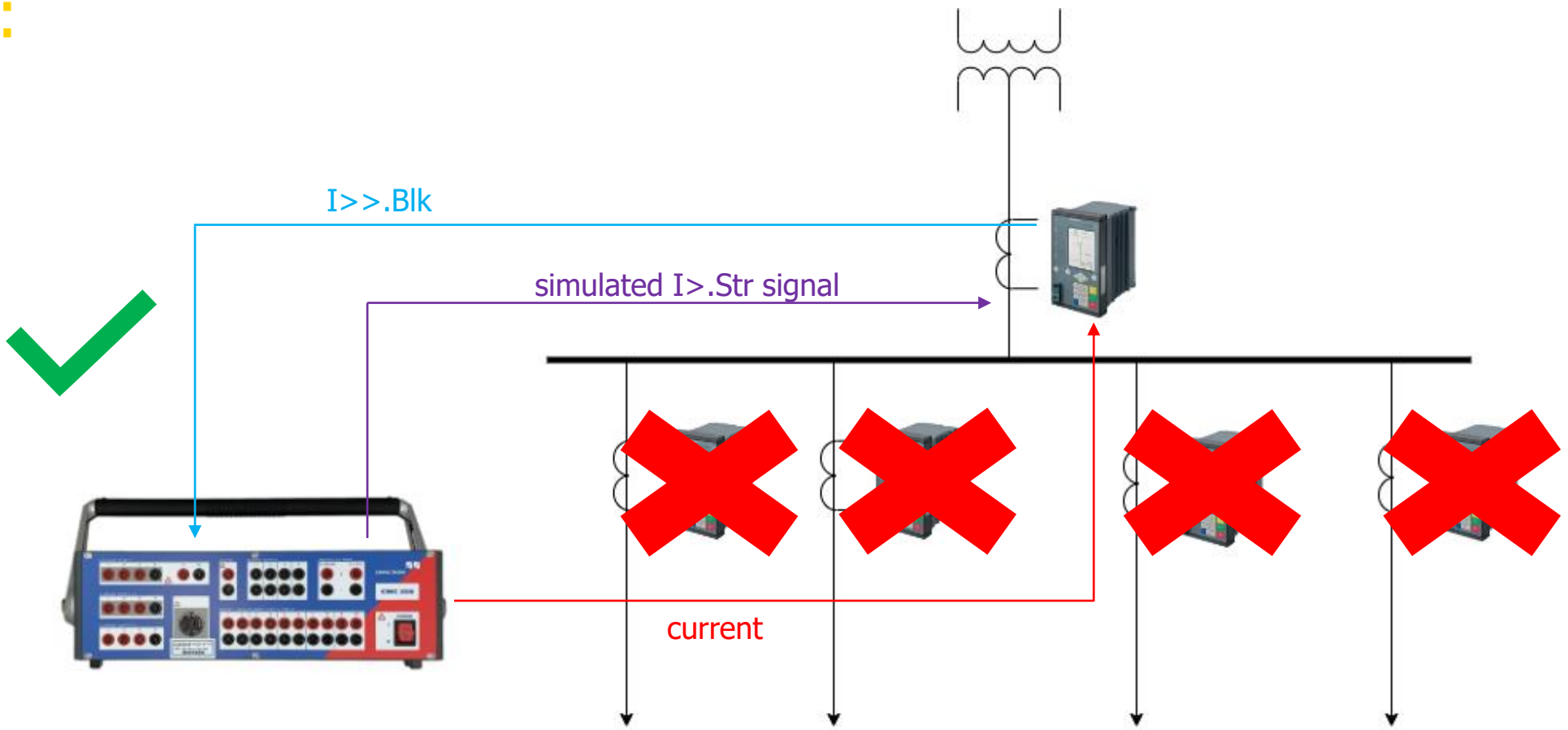
PART 1:



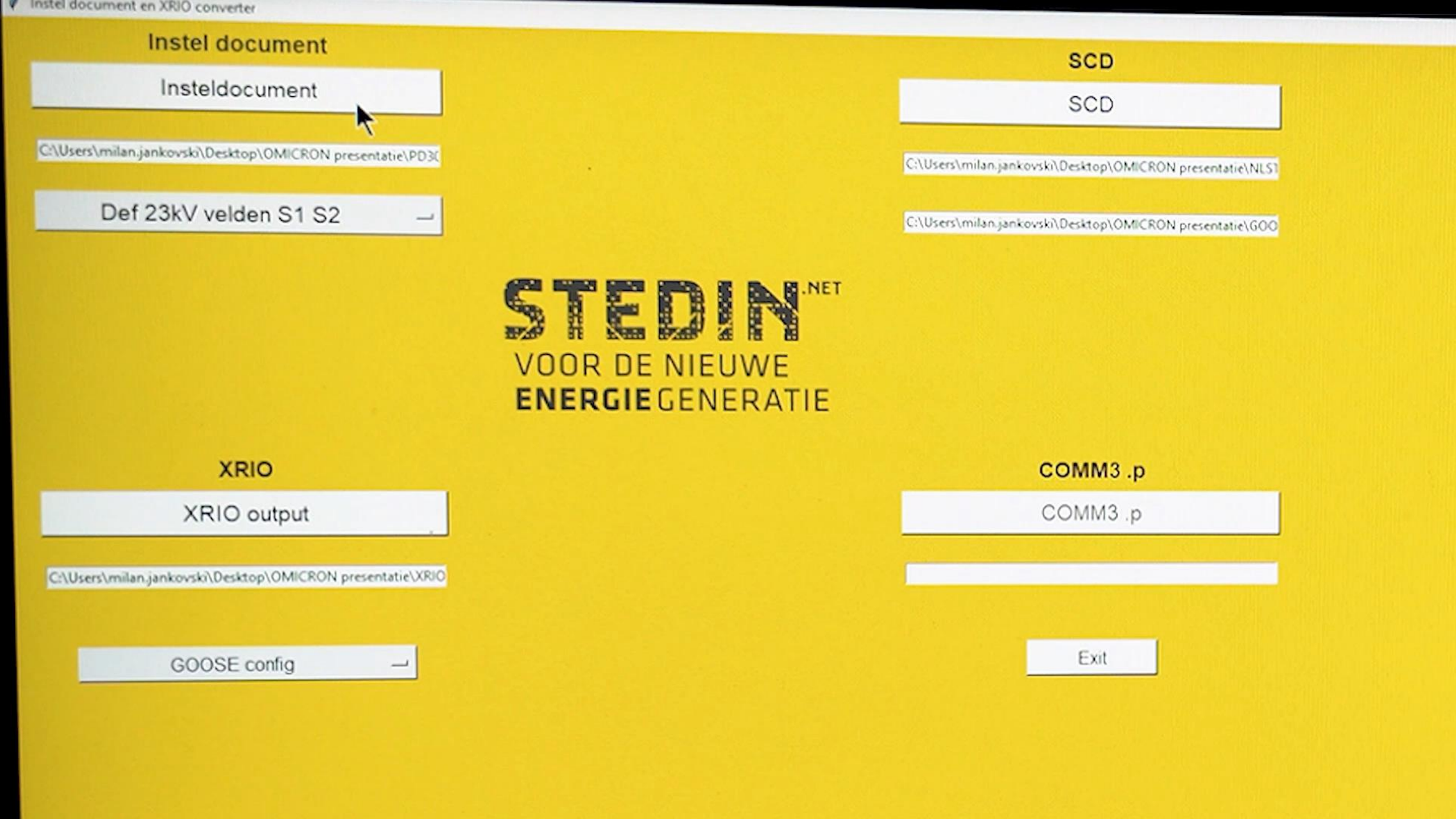
Example CASE - REVERSE-BLOCKING Scheme (3)

GOOSE-BASED TESTING

PART 2:



HOW DOES IT WORK IN PRACTICE


Instel document en XRIO converter
Instel document
Insteldocument
C:\Users\milan.jankovski\Desktop\OMICRON presentatie\PD3(
Def 23kV velden S1 S2
SCD
SCD
C:\Users\milan.jankovski\Desktop\OMICRON presentatie\NLS1
C:\Users\milan.jankovski\Desktop\OMICRON presentatie\GOO
STEDIN.NET
VOOR DE NIEUWE
ENERGIEGENERATIE
XRIO
XRIO output
C:\Users\milan.jankovski\Desktop\OMICRON presentatie\XRIO
GOOSE config
COMM3 .p
COMM3 .p
Exit

CONCLUSIONS

- GOOSE testing = less wiring = less time consuming
- Separated testing per function = more efficient testing
- Automated tool = less hand-filled data = less error-prone
- Python allows for easy and automated creation of OCC files
- Distributed protection schemes become easy to be tested

THANK YOU FOR YOUR ATTENTION