

**ZABBIX** '24

CONFERENCE

LATIN AMERICA

JUNE, 07-08, 2024. SÃO PAULO, BRAZIL

ZABBIX '24

CONFERENCE

LATIN AMERICA

# Explorando o Count: Novas perspectivas em pré-processamento

ZABBIX '24  
CONFERENCE

LATIN AMERICA

Bryam Balan

made  it



Linux  
Professional  
Institute

LPIC-1

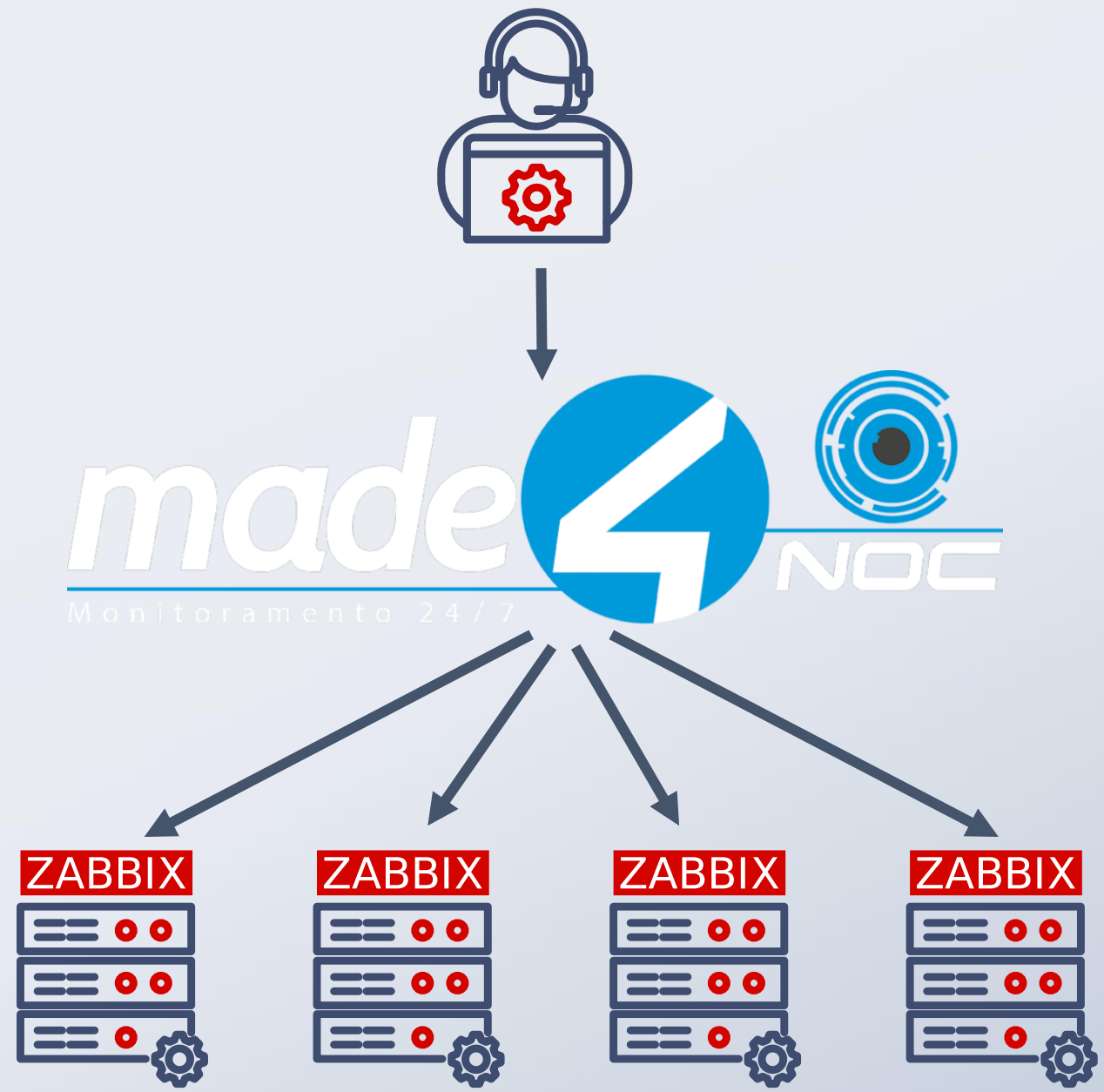


Linux  
Professional  
Institute

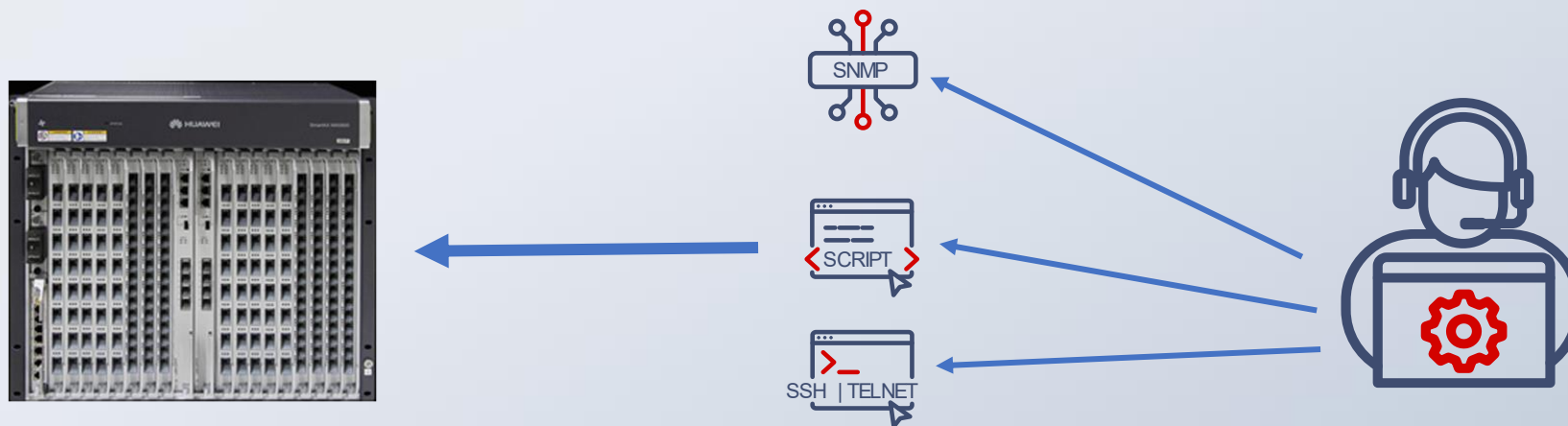
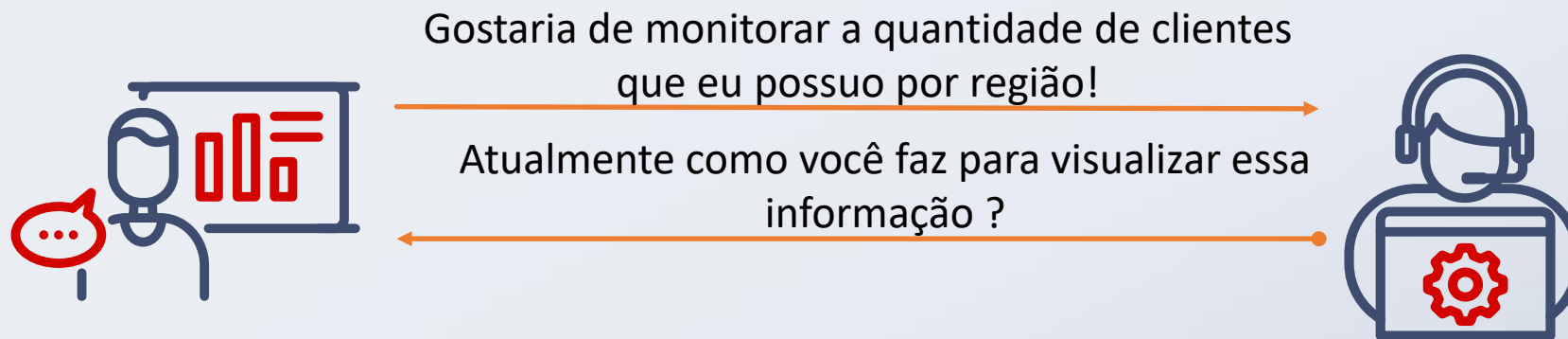
LPIC-2







# Como eu desenvolvia os monitoramentos



# Como eu desenvolvia os monitoramentos



# Requisição do Cliente



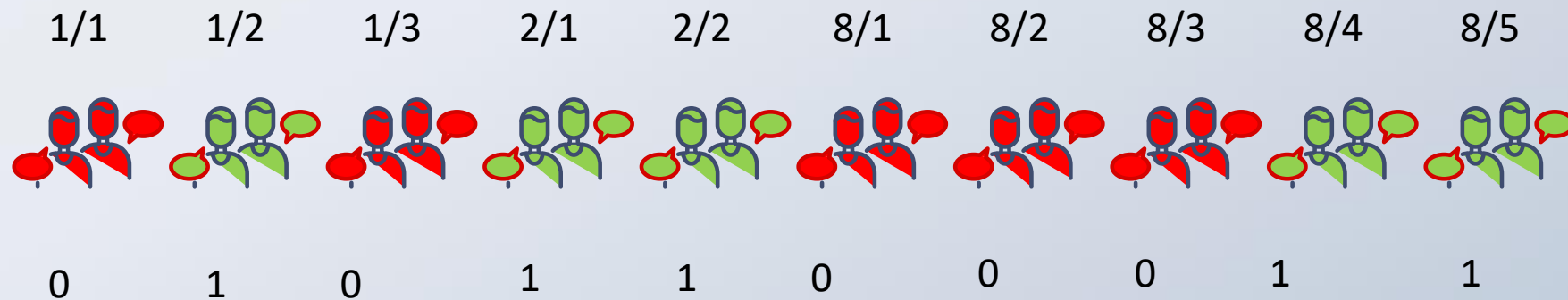
1/1	1/2	1/3	2/1	2/2	8/1	8/2	8/3	8/4	8/5
0	1	0	1	1	0	0	0	1	1
Região 1			Região 2		Região 8				

# Requisição do Cliente



## Objetivo:

- **Região 1**
  - Online: 1
  - Offline: 2
  - Total: 3
- **Região 2**
  - Online: 2
  - Offline: 0
  - Total: 3
- **Região 8**
  - Online: 2
  - Offline: 3
  - Total: 5

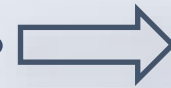
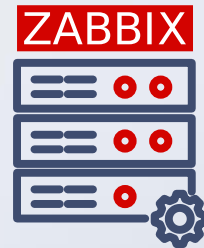
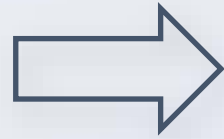




# Requisição do Cliente



Item	Valor
1/1	0
1/2	1
1/3	0
2/1	1
2/2	1
8/1	0
8/2	0
8/3	0
8/4	1
8/5	1



## Objetivo:

- **Região 1**
  - Online: 1
  - Offline: 2
  - Total: 3
- **Região 2**
  - Online: 2
  - Offline: 0
  - Total: 3
- **Região 8**
  - Online: 2
  - Offline: 3
  - Total: 5

# Requisição do Client

```
#!/bin/bash
```

```
URL='http://127.0.0.1/zabbix/api_jsonrpc.php'  
HEADER='Content-Type: application/json-rpc'  
HOST=$1  
PON=$2  
STATUS=$3
```

```
USER=robson  
PASS=jorgelaflon
```

```
autenticacao(){
```

```
    JSON='  
    {  
        "jsonrpc": "2.0",  
        "method": "user.login",  
        "params": {  
            "user": '$USER',  
            "password": '$PASS'  
        },  
        "id": 0
```

```
    }'  
    curl -s -X POST -H "$HEADER" -d "$JSON" "$URL" | cut -d '"' -f8
```

```
    }  
    TOKEN=$(autenticacao)
```

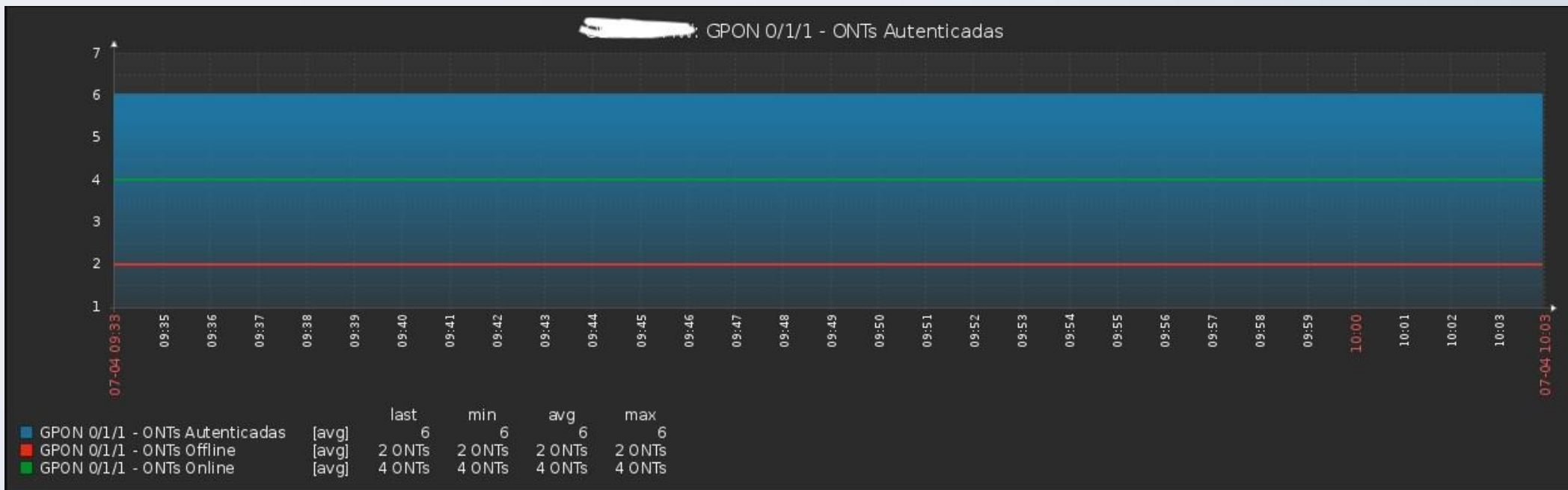
```
pegaritem()  
{
```

```
    JSON='  
    {  
        "jsonrpc": "2.0",  
        "method": "item.get",  
        "params": {  
            "output": ["lastvalue", "name", "key_", "snmp_oid"],  
            "host": "'$HOST'",  
            "search": {  
                "key_": "huawei.opticalPowerONU",  
                "snmp_oid": "'$PON'"  
            },  
            "sortfield": "name"  
        },  
        "auth": "'$TOKEN'",  
        "id": 1
```

```
    }'  
    curl -s -X POST -H "$HEADER" -d "$JSON" "$URL" | python -mjson.tool | grep -${STATUS} 21474836.4700 |  
    grep lastvalue | wc -l  
    }  
    pegaritem
```

# Requisição do Client

<input type="checkbox"/>	...	{#SNMPVALUE} - ONTs Autenticadas	ont-aut[{#SNMPVALUE}]
<input type="checkbox"/>	...	{#SNMPVALUE} - ONTs Offline	onu-status.sh["{HOST.NAME}", "{#SNMPINDEX}", "w"]
<input type="checkbox"/>	...	{#SNMPVALUE} - ONTs Online	onu-status.sh["{HOST.NAME}", "{#SNMPINDEX}", "v"]



# Requisição do Client

Manter o script funcional era uma dor de cabeça, compartilhar então...

## Requisitos:

- ⚡ Utilização do Shell Script
- ⚡ Ter Itens SNMP Criados no Zabbix
- ⚡ Ter acesso a API do Zabbix

## Possíveis Problemas:

- ⚡ Alteração do nome da chave do item
- ⚡ Alterações da chamadas da API do Zabbix



# Como o count me ajuda a simplificar tudo isso

O que é um Pré Processamento ?  
Um Criador/Transformador de dados

Sabem quantos pré processamentos existem ??

Items

All hosts / Zabbix server Enabled ZBX Items 145 Triggers 75 Graphs 28 Discovery rules 3 Web scenarios

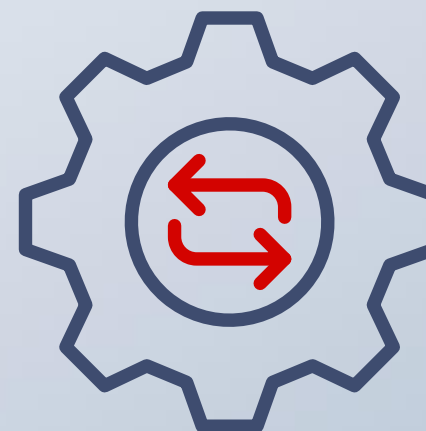
Item Tags Preprocessing 2

Preprocessing steps	Name	Parameters	Custom on fail
1:	Change per second		<input type="checkbox"/>
2:	Custom multiplier	0.01	<input type="checkbox"/>

[Add](#)

Type of information Numeric (float)

[Add](#) [Test](#) [Cancel](#)





# Como o count me ajuda a simplificar tudo isso

**Texto**

- Expressão regular
- Substituir
- Remover espaços
- Remover espaços à direita
- Remover espaços à esquerda

**Dados estruturados**

- XML XPath
- JSONPath
- CSV para JSON
- XML para JSON

**SNMP**

- Valor de caminhada SNMP
- SNMP walk to JSON

**Aritmética**

- Multiplicador

**Modificar**

- Alteração simples
- Modificações por segundo

**Sistema numeral**

- Lógico para decimal
- Octal para decimal
- Hexadecimal para decimal

**Scripts personalizados**

- JavaScript**

**Validação**

- No Intervalo
- Corresponde à expressão regular
- Não corresponde à expressão regular

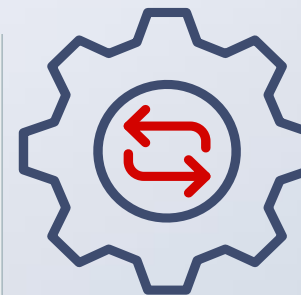
- Verificar se há erro no JSON**
- Verificação de erros no XML
- Verificação de erro usando expressão re...
- Verificação de valor não suportado

**Limitação**

- Descartar inalterado
- Descartar inalterado com heartbeat

**Prometeu**

- Padrão Prometheus
- Prometheus para JSON



# Como o count me ajuda a simplificar tudo isso

## O que é o count ?

Basicamente, minha manifestação para a utilização de **PIPE**  
**GREP PIPE WC -l**

Resumidamente, filtrar uma informação e contar quantas vezes ela apareceu, muito semelhante ao log.count.

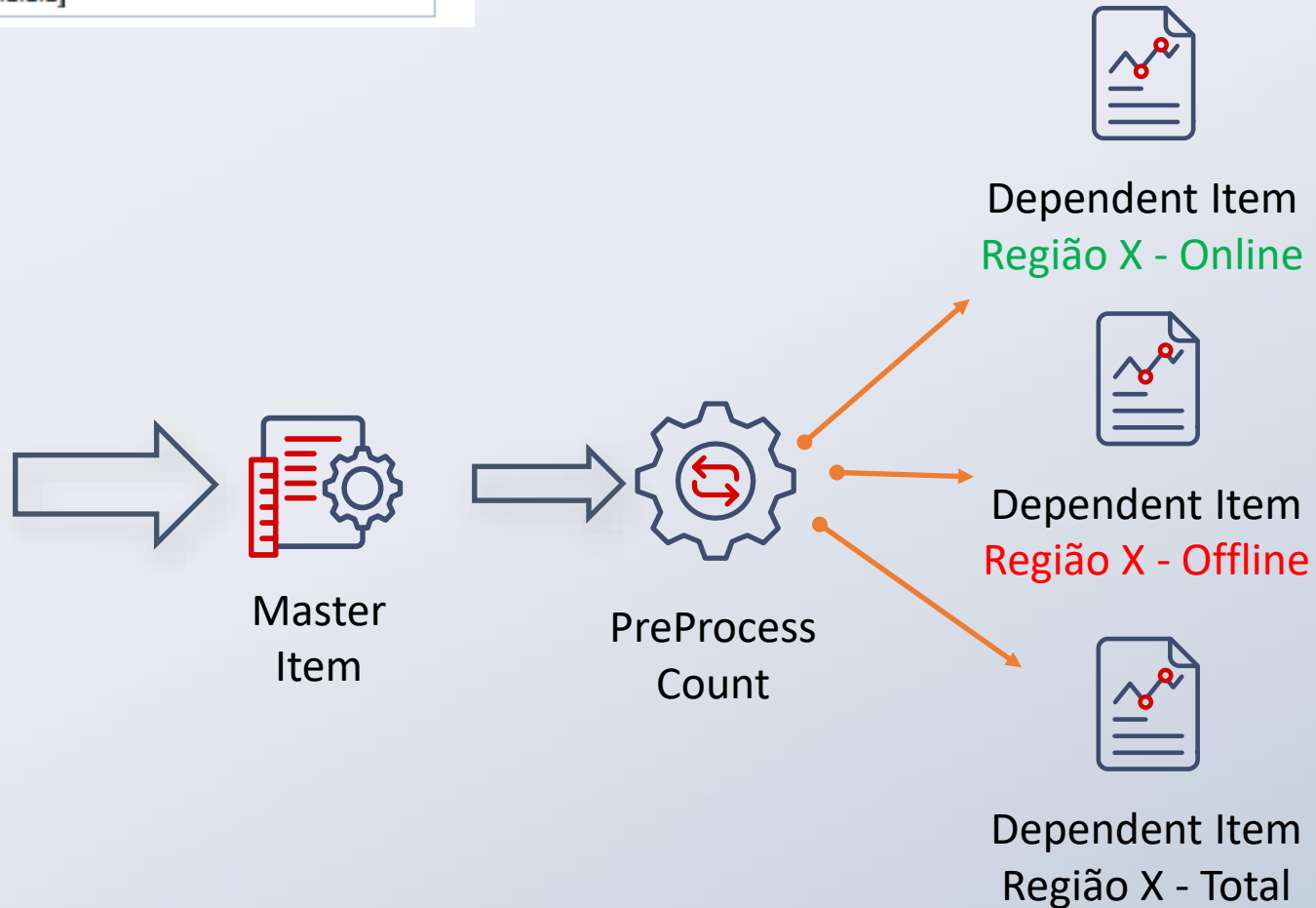


```
dante@Robson:~$ cat /etc/passwd | grep nologin | wc -l  
39
```


# Como o count me ajuda a simplificar tudo isso

\* SNMP OID

Item	Valor
1/1	0
1/2	1
1/3	0
2/1	1
2/2	1
8/1	0
8/2	0
8/3	0
8/4	1
8/5	1



# Como o count me ajuda a simplificar tudo isso

Por baixo dos panos é um simples JavaScript! 

```
function (value) {  
  var lines = value.split('\n');  
  var filteredLines = lines.filter(function (line) {  
    return line.includes("ITEM XYZ") && line.includes("VALOR XYZ");  
  });  
  var result = filteredLines.length;  
  return result !== undefined ? result : 0;  
}
```

# Como o count me ajuda a simplificar tudo isso

## Alterações:

- Não precisa mais de Shell Script
- Não precisa mais de acesso a API
- **Não precisa mais criar um item por cliente**
- Itens somente do que o cliente precisa monitorar
- Processamento todo por conta do Zabbix
- Evolução da ferramenta, só melhora o monitoramento (chave walk[] e SNMP Async)



# Conscientização



Por que utilizar formas nativas/desenvolver na ferramenta ?



Facilidade para  
compartilhar



Suporte por conta da  
ferramenta!



Facilidade para  
atualizações

ZABBIX '24

CONFERENCE

LATIN AMERICA

Gracias!

