

The background of the banner features a network diagram with red lines and nodes on the left side, transitioning into a teal background with a bokeh effect of blue and white circles on the right side.

# ZABBIX 2022 Conference BRAZIL

ZABBIX 2022  
Conference  
BRAZIL

Problemas en que nos vimos envueltos y  
como nos desenvolvimos



# CUSTOS

MONITORING



# ¿Quiénes somos?

**Empresa con foco en el monitoreo**

**Especializada en Zabbix**

**Representantes oficiales Zabbix en Uruguay**

**Certified Partner**

**Training Partner**

**Comprometida con nuestros clientes**



¿Qué no somos?

**Expertos en todo**

A nuestro pesar, pero trabajando en ello



# Problemas reales



- **Comunicación**
  - La solicitud no expresa la necesidad
  - La solicitud no soluciona el problema
  - No entendemos la solicitud

- **Desconocimiento**
  - De la herramienta de terceros
  - De lo que Zabbix puede hacer
  - De las políticas del negocio que estamos monitoreando



# Cosechar datos cada 40 minutos

- Update Interval

- 40m

- Pero no tengo el valor a las 00:00



- Intervalos personalizados

- Soluciones

- m0/40 

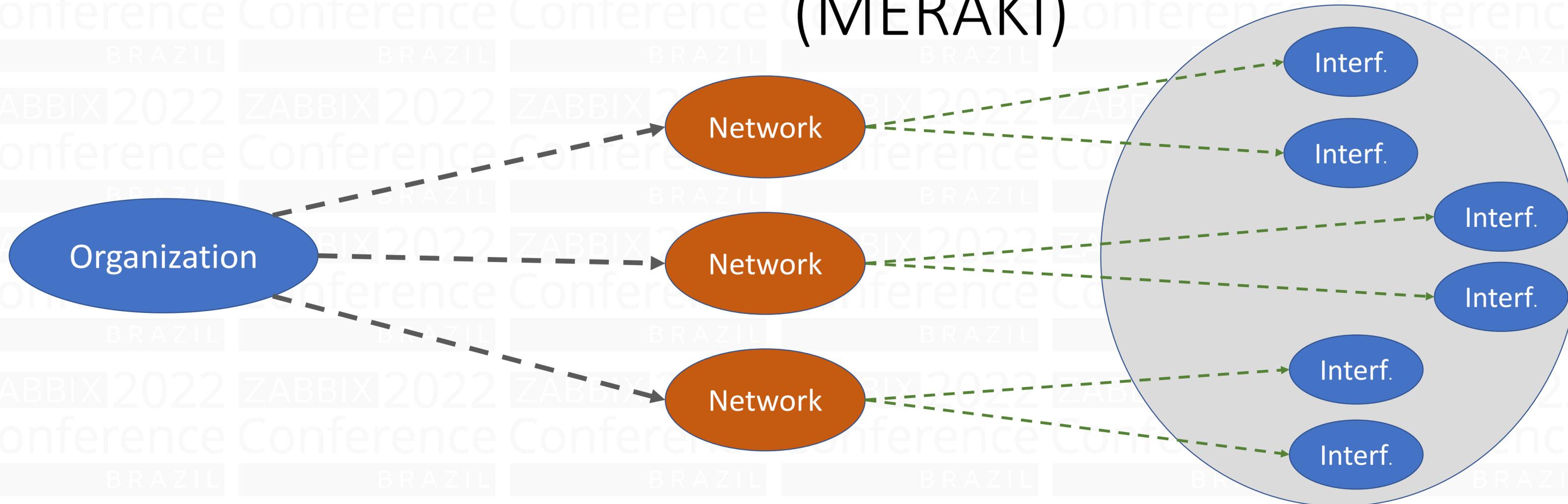
- Falla equivalente a m0,40

- h0-23/2m0/40

- h1-23/2m20



# No me avises de las interfaces si la red no está online (MERAKEI)



# Muchas interfaces en las distintas redes (LLD)

- Estado de las interfaces

- Llamado a la API

- Estado (active/ready ...)

- IP

- DHCP/ Static

- Network ID

- Otra Info irrelevante

- Trigger cuando

- ni active ni ready

- DHCP



# Muchas redes – sucursales (LLD)

- Estado de las Networks

- Llamado a la API

- Network id

- Estado de la network (online/no online)

- Otra info irrelevante

- Trigger cuando

- Cuando no online



# Dependencias entre triggers de distintas LLD

- Mezclar las LLD

- En el preproceso de una LLD
  - HTTPRequest a otra
  - Armado del nuevo JSON
    - Se arma el JSON con la info de los dos
    - Nueva LLD con los datos Finales



- Primer intento

- [ { DatosNetwork1, Crear, DatosInterface1 },  
{ DatosNetwork1, Crear, DatosInterface2 },  
{ DatosNetwork2, Crear, DatosInterface1 },  
{ DatosNetwork2, Crear, DatosInterface2 },  
...  
]



# Dependencias entre triggers de distintas LLD

- Nuevo problema

- La dependencia entre triggers solo la podemos definir dentro de la misma entidad



- Resultado final del JSON

```
[ { DatosNetwork1,  
  DatosInterface1,  
  DatosInterface2 },  
  { DatosNetwork2,  
    DatosInterface1,  
    DatosInterface2 },  
  ...  
]
```

# A seguir avanzando en el desarrollo (o el aprendizaje)

- Problema no resuelto (aún)
  - Las distintas networks tiene distinta cantidad de interfaces



- Work arround
  - Definir tantos ítems prototipes como el que más interfaces tiene
  - En la versión 5.2
    - No podíamos no especificar que la macro no existiera (si desde la 5.4)
    - Rellenamos todas la LLD Macros con valores dummy

# Deseos para el desarrollo futuro

- Oh! Zabbix escúchanos
  - Poder especificar una propiedad repetitiva en una entidad
    - Para crear en este caso las distintas interfaces
    - Para crear distintas tags en los ítems y triggers
  - <https://support.zabbix.com/browse/ZBXNEXT-1527>



# Sirena frente a alertas (IoT)

- Activar una sirena frente a alertas
  - Si reaccionamos a cada alerta con la sirena, puede estar sonando insistentemente



- Host Resumen
  - Script Item
    - Con la API sacamos la cantidad de triggers activos en grupos de hosts
    - Parámetros de la función mediante user macros
  - El Arduino consulta los nuevos eventos de este host

# Script



```
req = new HttpRequest()  
req.addHeader('Content-Type: application/json')
```

```
jdata = { "jsonrpc": "2.0",  
          "method": "trigger.get",  
          "params": { "filter": { "value": 1 },  
                    "groupids": [{$BUZZER.HOSTGROUPSIDS}],  
                    "min_severity": {$BUZZER.TRIGGER.MINSEVERITY},  
                    "skipDependent": {$BUZZER.TRIGGER.SKIPDEPENDENT},  
                    "countOutput": true,  
                    "tags" : [{$BUZZER.TRIGGER.TAG}]  
          },  
          "auth": "{$ZBX.USER.TOKEN}",  
          "id": 1  
        }
```

```
return JSON.parse( req.get( '{$ZBX.URL}', JSON.stringify( jdata ) ) ).result
```

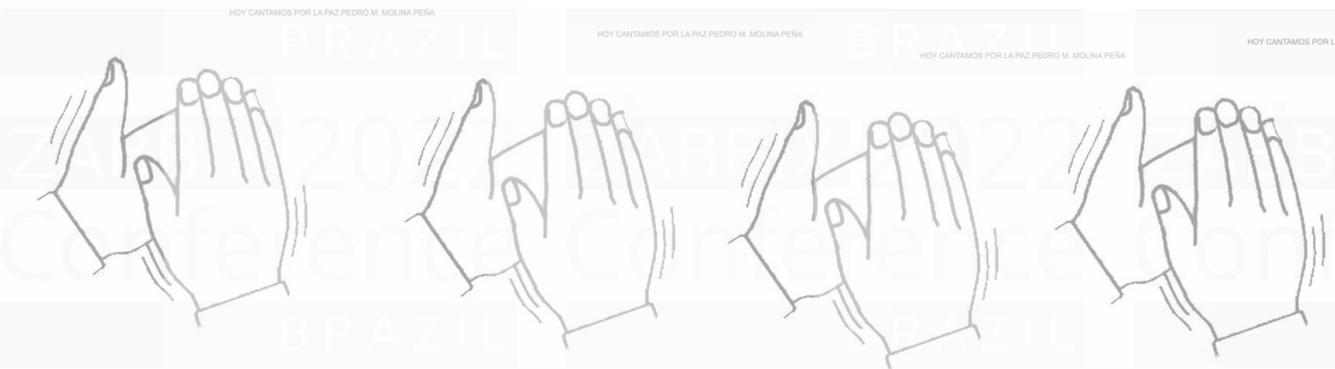
# Control de sonido y repetición de la alerta

- Frecuencia
  - Controlando el update Interval del item
- Repetición
  - Múltiple o single evento en el trigger
- Diferentes sonidos para diferentes triggers
- Codificación en la descripción del trigger
  - PROBLEM PIN 13 HIGH 200
  - PROBLEM PIN 13 LOW 100
  - PROBLEM PIN 13 HIGH 200
  - PROBLEM PIN 13 LOW 100
  - OK PING 13 HIGH 500
  - OK PING 13 LOW 300
  - OK PING 13 HIGH 500
  - OK PING 13 LOW 300



# Reintentar frente a unsupported

- Update Interval muy grande
  - Control de ingresos de pedidos
    - Query a una DB
      - Time out de la base al ejecutar
  - El query se hace una vez por día
  - El time out se da por la exigencia que tiene la DB en un momento
- Reintento automático
  - Internal Action
    - Creamos una Tarea Execute Now



# Las posibilidades

- Definimos un script Action Operation
  - HTTPRequest a la API de Zabbix y cree un Task execute now
  - Internal Action que ejecute
    - No es posible seleccionar un script ❌
- Preproceso del item
  - Chequeo for non supported
    - Ponemos un valor bandera
  - Trigger a valor bandera
    - Action del trigger que llame al script



# Script



```
params = JSON.parse(value)

try {
  req = new HttpRequest()
  req.addHeader('Content-Type: application/json')
  SesID = params["token"]

  jdata = {"jsonrpc": "2.0",
          "method": "task.create",
          "params": [ {"type": "6",
                      "request": { "itemid": params["ItemID"] }
                    ]
          },
          "auth": SesID,
          "id": 1
        }

  Zabbix.log(3, "Reintento automatico:" +
              req.get( params["URL"], JSON.stringify( jdata ) ) )
} catch (error) { Zabbix.log(3, "Reintento automatico ERR " + error) }
```

# Controles

- En el action definimos los steps
  - En cada step se llama al script
    - Cuando se tenga éxito
      - se apaga este trigger
      - se detiene el escalado
      - se obtuvo el valor
      - se diparan los triggers correspondientes
- Controlamos qué tan insistentes nos ponemos
  - Por ejemplo
    - Steps 2 – 0  
Step Interval 60s
    - Steps 2 – 5  
Step Interval 5m  
Steps 5 – 0  
Step Interval 60s



ZABBIX 2022  
Conference  
BRAZIL

Muchas gracias!



# CUSTOS

MONITORING

<https://custos.uy>

[info@custos.uy](mailto:info@custos.uy)

<https://t.me/zabbixespanol>