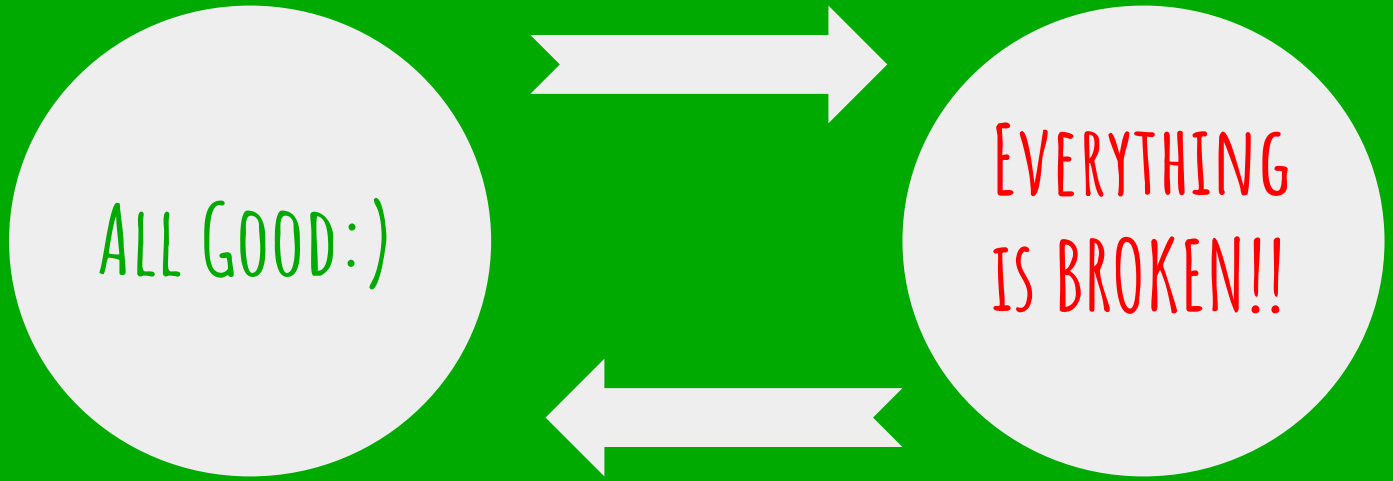
Simple Network Monitoring

Victor Zakharyev
Workday

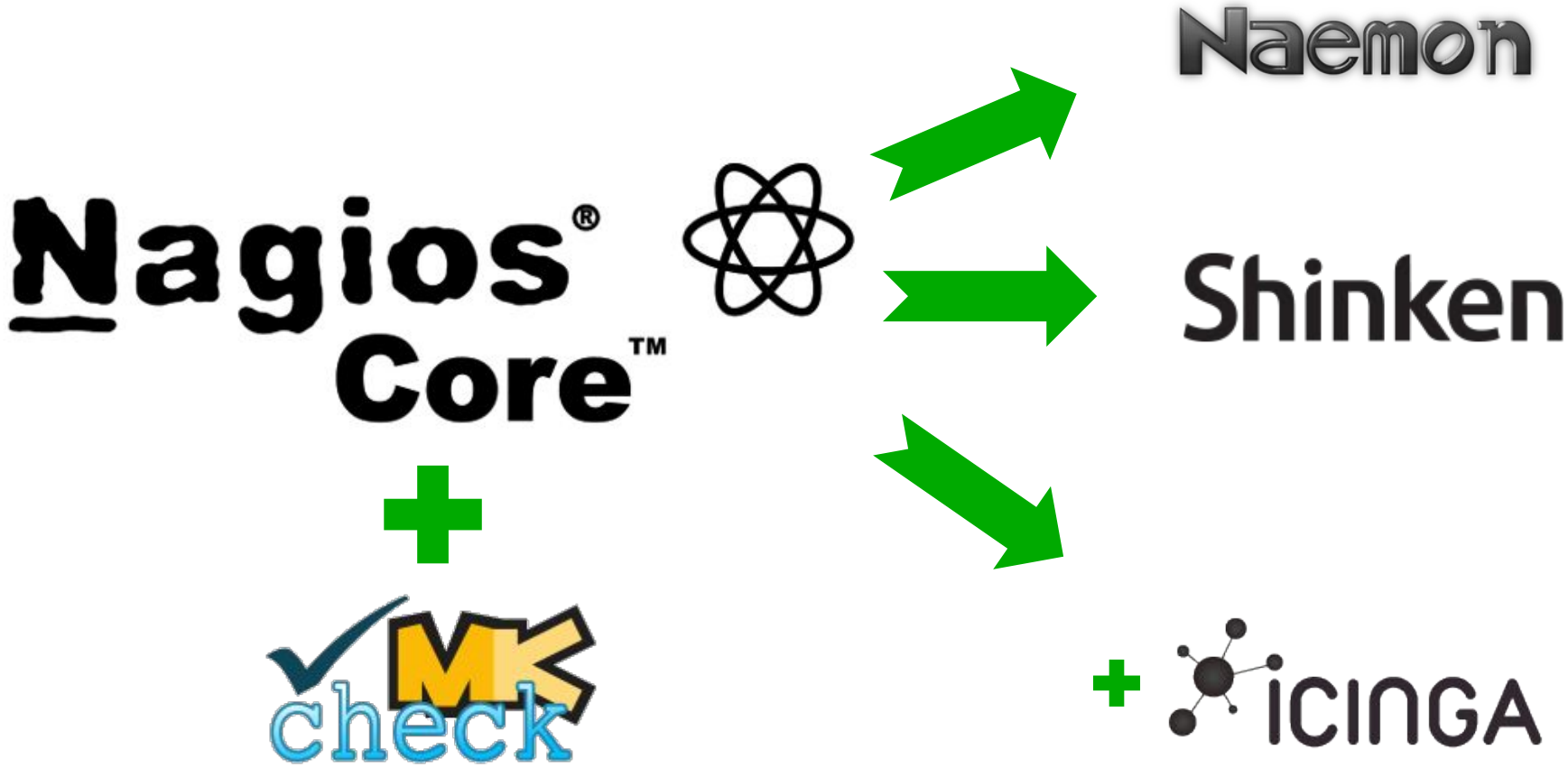
- Time Series
- Events
- State <<< !!!

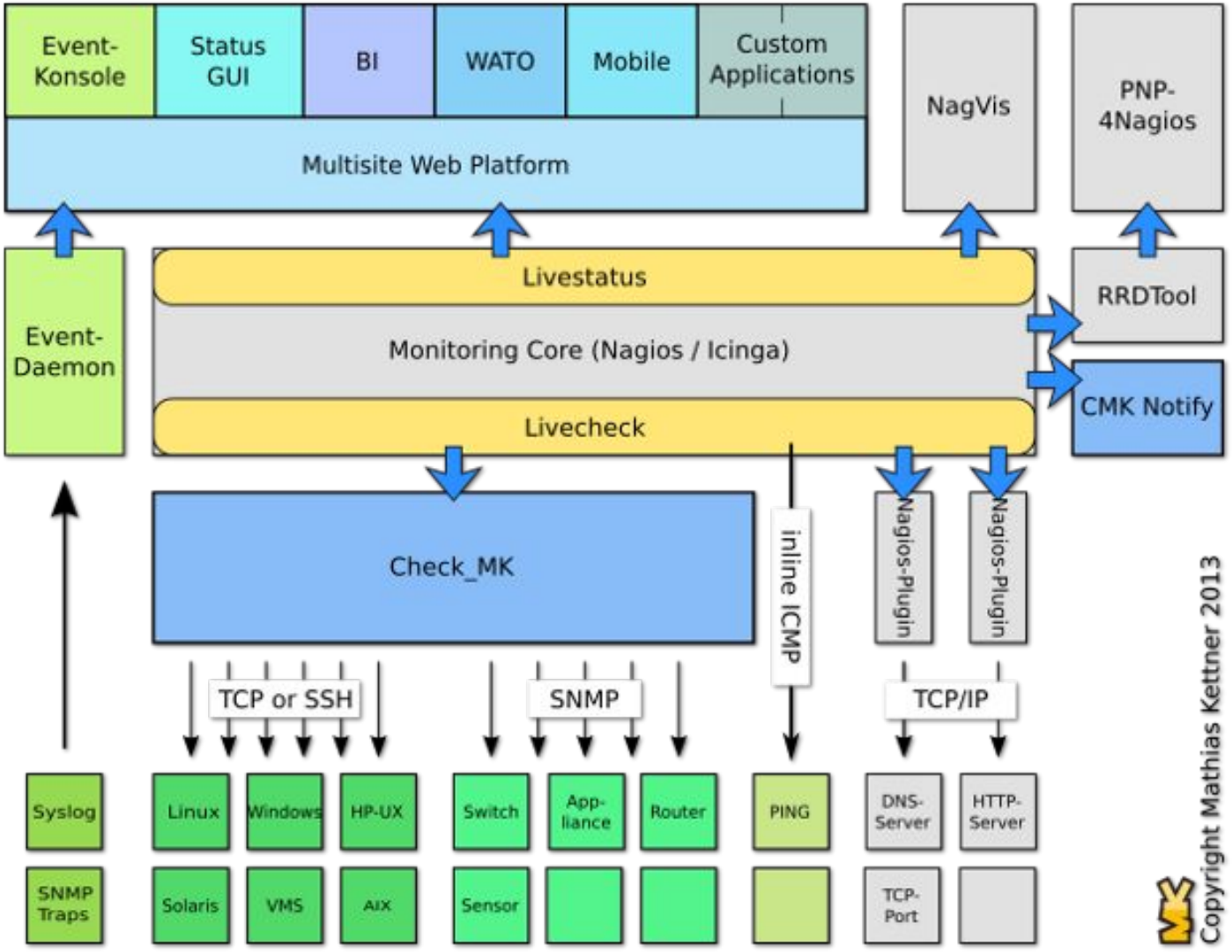

All Good :)
Everything is BROKEN!!

Simple Network Management Protocol

(not really)

(...right)


Naemon
Nagios®
Core™
Shinken
+
check
MK
+
ICINGA


Event-Konsole
Status GUI
BI
WATO
Mobile
Custom Applications
Multisite Web Platform
NagVis
PNP-4Nagios
Event-Daemon
Livestatus
Monitoring Core (Nagios / Icinga)
Livecheck
RRDTool
CMK Notify
Check_MK
inline ICMP
Nagios-Plugin
Nagios-Plugin
TCP or SSH
SNMP
TCP/IP
Syslog
Linux
Windows
HP-UX
Switch
App-liance
Router
PING
DNS-Server
HTTP-Server
SNMP Traps
Solaris
VMS
AIX
Sensor
TCP-Port
Copyright Mathias Kettner 2013


FORTINET
HEWLETT-PACKARD
TP-LINK
AEROHIVE NETWORKING
DELL
ADVA OPTICAL NETWORKING
INNOVAPHONE
ARBOR
BLUECAT
ENTERASYS
CISCO
EXTREME NETWORK
F5
KEMP
BLUE COAT
JUNIPER
PALO ALTO
INTEL
FIREEYE
INFOBLOX
CBL
PAN DACOM
BROCADE
ARRIS
VIPRINET
PLUGINS
GENUA
GENERIC CHECK
NETGEAR
BINTEC
CHECKPOINT
MIKROTIK

Okey, check that I need
is still not there,
how can I add it?

Specific MIB Information

Name	isisISAdjState
OID	1.3.6.1.2.1.138.1.6.1.1.2
Syntax	INTEGER down(1) initializing(2) up(3) failed(4)
Access	read-only
Status	current
Description	The state of the adjacency.
Reference	{ISIS.aoi adjacencyState (78)}
MIB	ISIS-MIB

MIB Hierarchy

iso → org → dod → internet → mgmt → mib-2 → isisMIB → isisObjects → isisISAdj →

- isisISAdjTable
  - isisISAdjEntry
    - isisISAdjIndex
    - **isisISAdjState**
    - isisISAdj3WayState
    - isisISAdjNeighSNPAAddress
    - isisISAdjNeighSysType
    - isisISAdjNeighSysID
    - isisISAdjNbrExtendedCircID
    - isisISAdjUsage
    - isisISAdjHoldTimer

Some initial code to get started

```
def inventory_mynics(info):
   # Debug: lets see how the data we get looks like
   print info
   return []

def check_mynics(item, params, info):
   return (3, "UNKNOWN - not yet implemented")

check_info["mynics"] = {
    "check_function"         : check_mynics,
    "inventory_function"     : inventory_mynics,
    "service_description"    : "NIC %s",
    "snmp_info"              : ( ".1.3.6.1.2.1.2.2.1", [ "2", "3", "8" ] )
}
```

Slowly fill some details

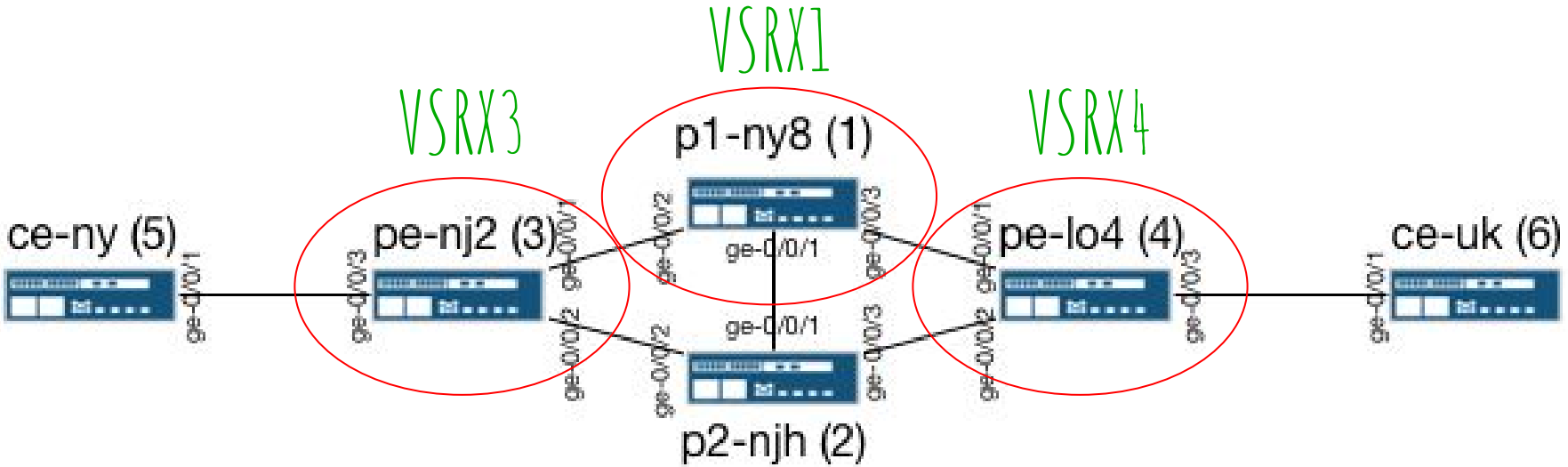
```
def inventory_mynics(info):
   for nic, type, state in info:
       if type == "6" and state == "1":
      yield, nic, None

def check_mynics(item, _no_params, info):
   for nic, type, state in info:
      if nic == item:
     if state == "1":
        return 0, "OK - link is up"  # All Good :)
     else:
        return 2, "CRITICAL - link is " + state  #Everything is BROKEN!!11

check_info["mynics"] = {
    "check_function"         : check_mynics,
    "inventory_function"     : inventory_mynics,
    "service_description"    : "NIC %s",
    "snmp_info"              : ( ".1.3.6.1.2.1.2.2.1", [ "2", "3", "8" ] )
}
```

Demo time...

SP Lab layout



Links and additional Info

- Check_MK main web-site:
  http://mathias-kettner.com/check_mk.html
- Check_MK SNMP Plugin tutorial:
  http://mathias-kettner.com/checkmk_devel_snmpbased.html
- LAB's monitoring box:
  https://github.com/vazic/checkmk-vagrant
- LAB's network devices:
  https://github.com/vazic/inex_meeting