

Software-Defined Data Center *und Monitoring*

19.05.2013 | Workshop Hamburg

 **NETWAYS**®

Bernd Erk
www.netways.de

VORSTELLUNG NETWAYS

EINFÜHRUNG

DEVOPS

KOMPONENTEN

MOTIVATION

TECHNISCHER ÜBERBLICK

MONITORING

ZUSAMMENFASSUNG

F&A





VORSTELLUNG NETWAYS

NETWAYS Kompetenzen

OPEN SOURCE SYSTEMS MANAGEMENT

- Monitoring & Reporting
- Configuration Management
- Service Management
- Knowledge Management
- Backup & Recovery

OPEN SOURCE DATA CENTER

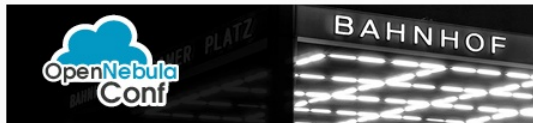
- High Availability & Clustering
- Cloud Computing
- Load Balancing
- Virtualization
- Database Management

MANAGED SERVICES

MONITORING HARDWARE

KONFERENZEN

NETWAYS Konferenzen



OpenNebula Conf 2013

24 - 26 September in Berlin

OpenNebulaConference 2013 24 – 26 September 2013

- www.opennebula.conf
- Programm geht diese Woche online



Open Source Monitoring Conference 23 – 24 Oktober 2013

- 250 Teilnehmer (2012)
- Monitoring Best Practice
- Aktuell läuft noch der CFP



EINFÜHRUNG

Last Big Things

CLOUD

BIG DATA

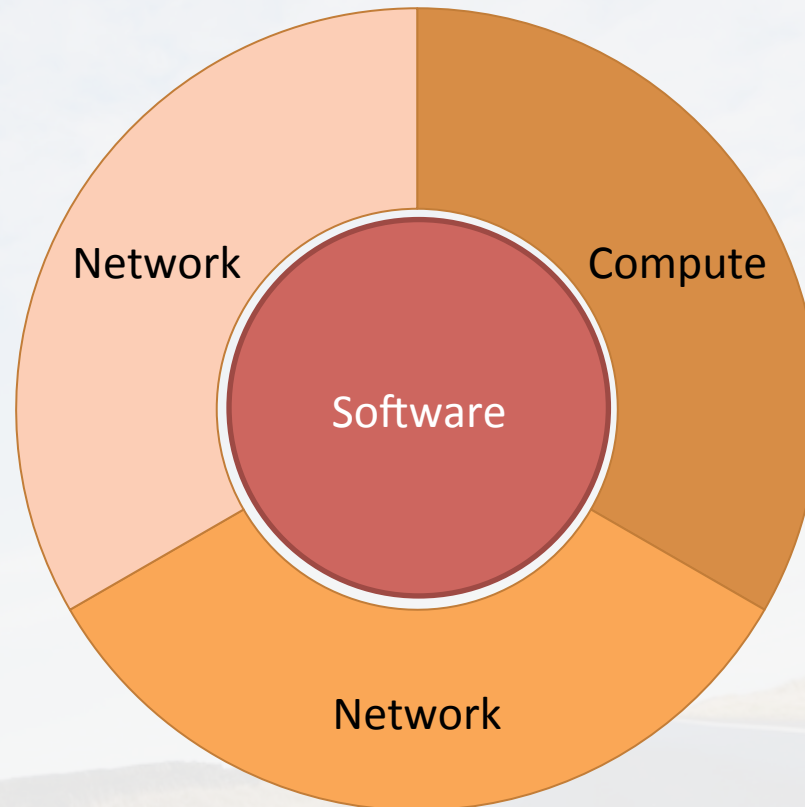
PRISM

Software-Defined Data Center

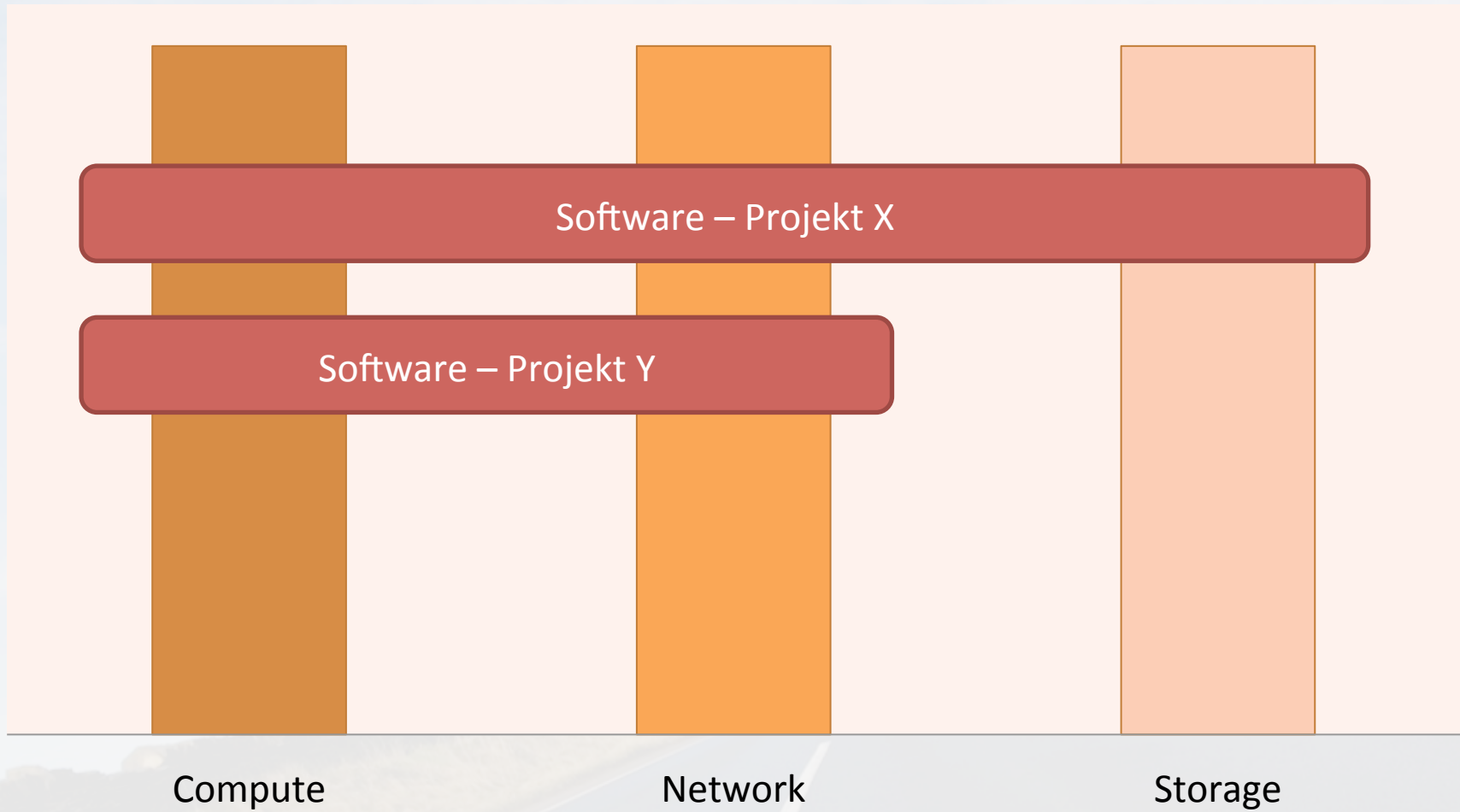
Software-Defined Data Center

SDDC

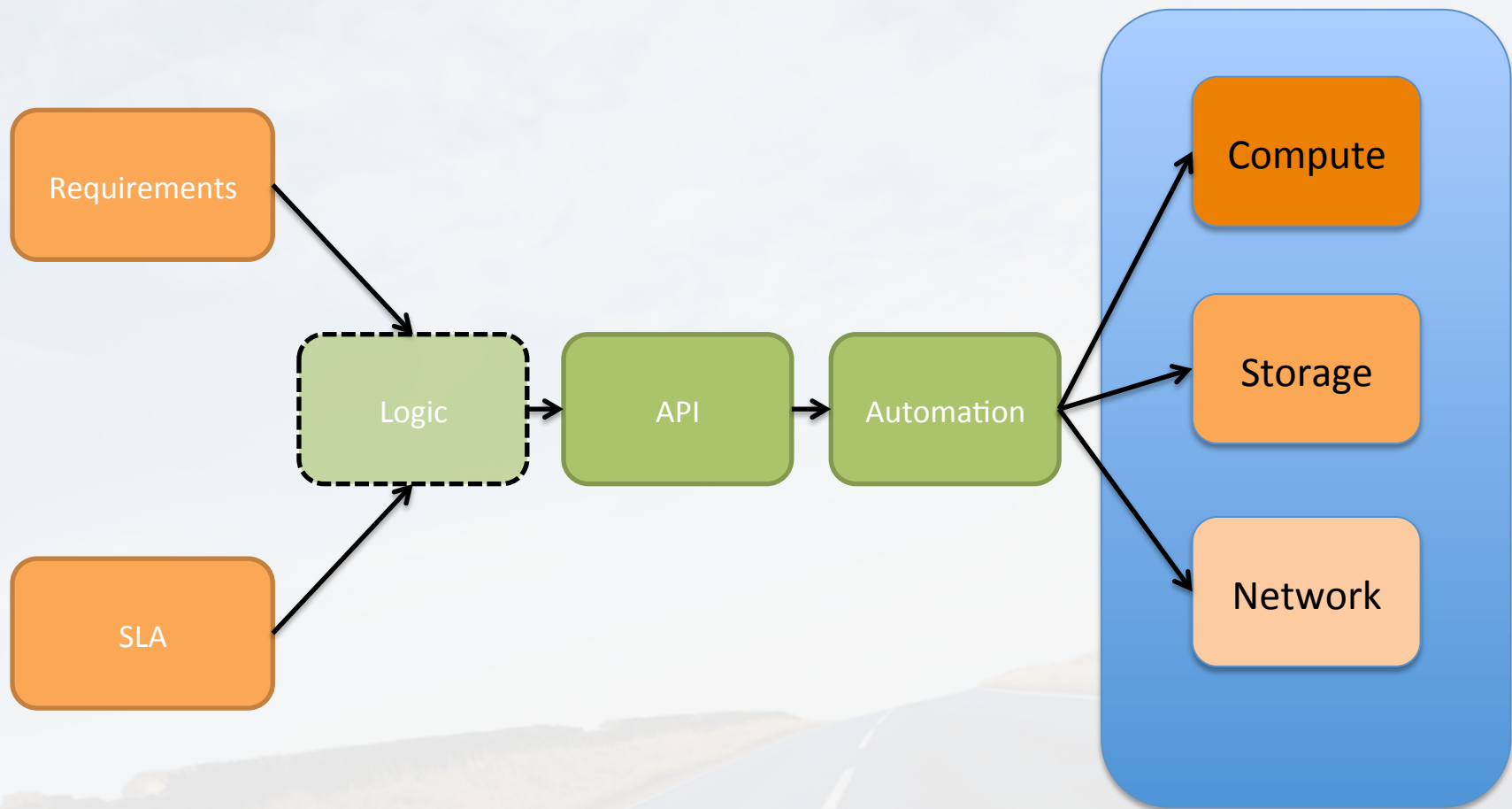
SDDC - Komponenten



SDDC - Idee



SDDC - Workflow



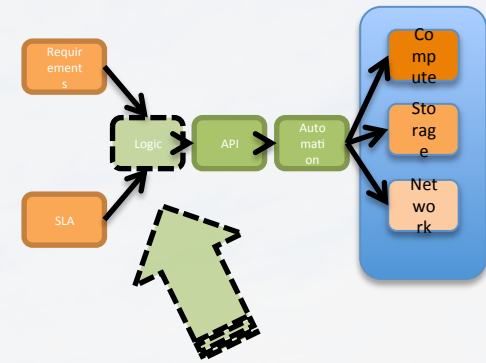
The logo for NETWAYS, featuring the word "NETWAYS" in a bold, white, sans-serif font. To the left of the text is a vertical orange bar with a white triangle pointing to the right at the top. A registered trademark symbol (®) is located to the right of the word.

NETWAYS®

DEVOPS

SDDC – Missing/Incomplete Piece

DevOps



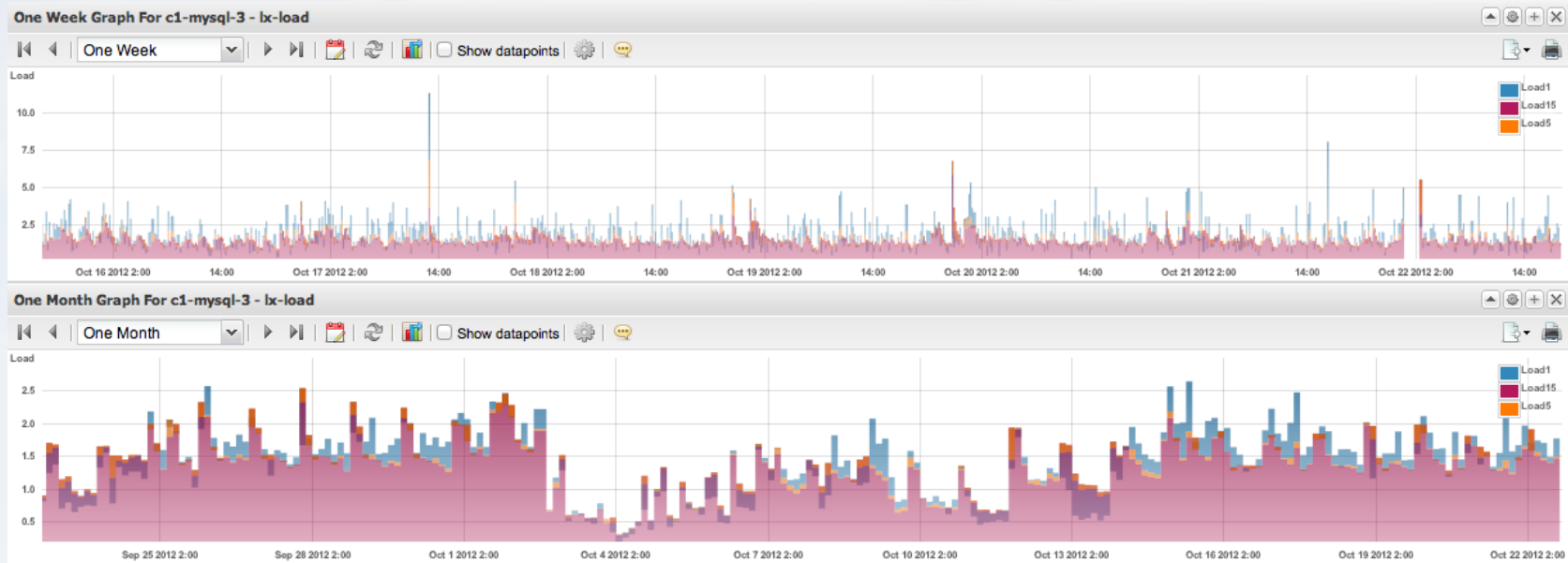
Was ist DevOps?

“DevOps is a cultural and professional movement”
(Adam Jacob)

Was ist DevOps?

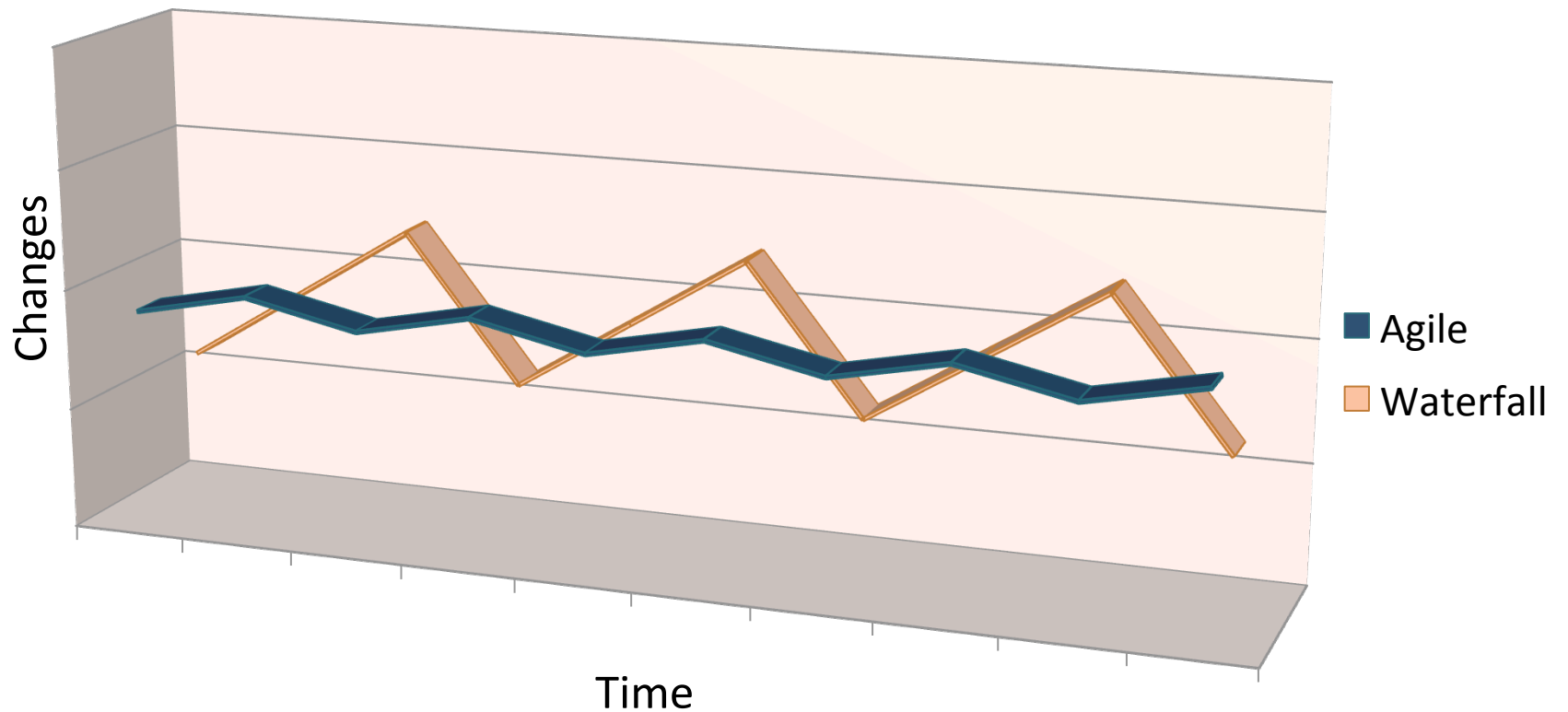


DevOps Stack - Monitoring



Wenn man es nicht messen kann,
kann es nicht wichtig sein!

DevOps Stack - Agilität



Was ist DevOps nicht?



DevOps Portal

The Oracle logo, consisting of the word "ORACLE" in red, bold, sans-serif capital letters. A registered trademark symbol (®) is located at the top right of the logo.

ORACLE®

DevOps Builder

NO HIPPO



NETWAYS[®]

HIghest Paid Persons Opinion

www.netways.de

Darum prüfe wer sich ewig bindet

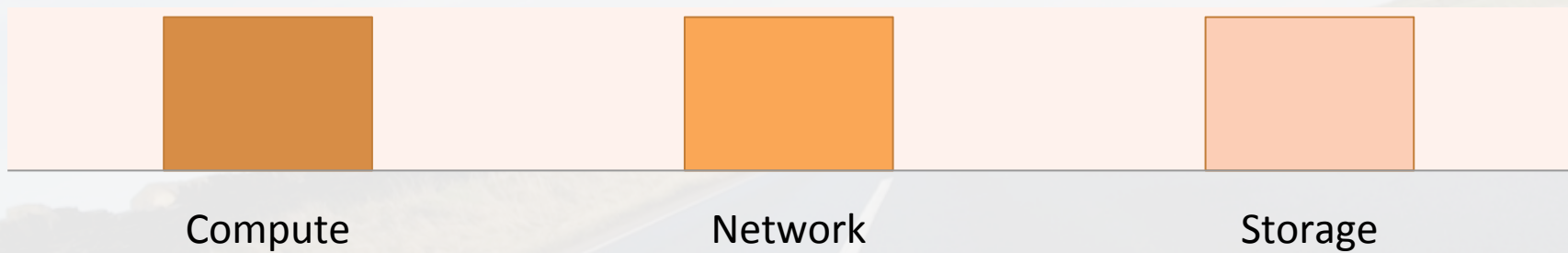




KOMPONENTEN

SDDC - Komponenten

- Vmware vCloud
- Redhat / HP
- OpenNebula
- CloudStack
- ...
- Nicira
- Open vSwitch
- OpenFlow
- Cisco
- Juniper
- ...
- Ceph
- GlusterFS
- NetApp
- EMC
- ...



SDDC – Automation und Orchestration

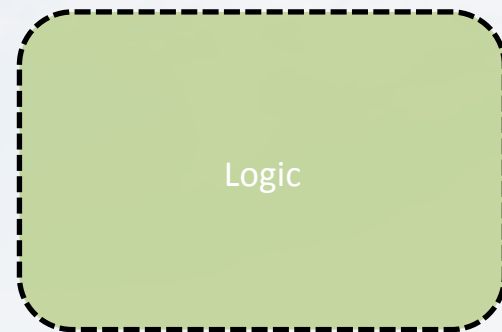
■ Open Source

- ▶ Puppet
- ▶ Chef
- ▶ cfengine
- ▶ Ansible
- ▶ (R)?ex

■ Kommerziell

- ▶ UC4
- ▶ KACE
- ▶ System Center Configuration Manager

Logic Stack



- Individuallösungen basierend auf
 - ▶ Puppet
 - ▶ Chef
 - ▶ Eigene Scripts
 - ▶ CMDB-Anbindungen
 - ▶ Bash oder ähnliches



NETWAYS®

MOTIVATION

Warum SDDC – Technisch

- Die Technologie die Möglichkeiten bieten
- Reine Virtualisierung an ihre Grenzen kommt
- Unterschiedliche Anforderungen innerhalb eines Datacenters erfüllt werden müssen
- Server, Netz und Storage auch in "kleineren" Firmen Standardleistungen werden

Warum SDDC – Marketing

- Warum nicht?
- Etwas Neues am Markt benötigt wird
- Mit Virtualisierung allein zu wenig Geld zu verdienen ist
- Die geänderten Anforderungen der Kunden eine Lösung benötigen

Warum SDDC – Fachlich

Service
Operation

Service
Design

Service
Transition



TECHNISCHER ÜBERBLICK

Erzeugen einer virtuellen Maschine - Puppet

```
$ puppet help node_vmware
USAGE: puppet node_vmware <action>
```

This subcommand provides a command line interface to work with VMware vSphere Virtual Machine instances. The goal of these actions is to easily create new virtual machines, install Puppet onto them, and clean up when they're no longer required.

OPTIONS:

```
--mode MODE           - The run mode to use (user, agent, or master).
--render-as FORMAT    - The rendering format to use.
--verbose              - Whether to log verbosely.
--debug               - Whether to log debug information.
```

ACTIONS:

```
create      Create a new VM from a template
find        Find a VMware Virtual Machine
list        List VMware Virtual Machines
start       Start a Virtual Machine
stop        Stop a running Virtual Machine
terminate   Terminate (destroy) a VM
```

See 'puppet man node_vmware' or 'man puppet-node_vmware' for full help

Erzeugen einer virtuellen Maschine - Puppet

```
$ puppet node_vmware create --name=newpuppetmaster --template="/
Datacenters/Solutions/vm/master_template"
notice: Connecting ...
notice: Connected to vc01.example.com as cloudprovisioner (API version
4.1)
notice: Locating VM at /Datacenters/Solutions/vm/master_template
(Started at 12:38:58 PM)
notice: Control will be returned to you in 10 minutes at 12:48 PM if
locating (1/2) is unfinished.
Locating (1/2):      100% |
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo| Time:
00:00:16
notice: Starting the clone process (Started at 12:39:15 PM)
notice: Control will be returned to you in 10 minutes at 12:49 PM if
starting (2/2) is unfinished.
Starting (2/2):      100% |
ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo| Time:
00:00:03
---
name: newpuppetmaster
power_state: poweredOff
...
status: success
```

Erzeugen einer virtuellen Maschine - Ansible

```
### Kick off HP iLO provisioning
- local_action: hpilo_boot host='${cmdb_parent}' login='${ilologin}'
password='${ilopassword}' media='cdrom' image='http://${ansible_server}/iso/
ks/${inventory_hostname_short}.iso' state='boot_once'
  only_if: "'${cmdb_hwtype}'.startswith('HP ')"

### Kick off vSphere provisioning
- local_action: vsphere_boot host='${cmdb_parent}' login='${esxlogin}'
password='${esxpassword}' guest='${cmdb_uuid}' media='cdrom' image='[nfs-
datastore] /iso/ks/${inventory_hostname_short}.iso' state='boot_once'
  only_if: "'${cmdb_hwtype}'.startswith('VMWare ')"

### Kick off KVM provisioning
- local_action: virt_boot guest='${cmdb_uuid}' media='cdrom' image='/iso/
ks/${inventory_hostname_short}.iso' state='boot_once'
  delegate_to: '${cmdb_parent}'
  only_if: "'${cmdb_hwtype}'.startswith('KVM')"
```

Konfiguration VLAN – Junos I

```
node "myswitch1234.mycorp.com" {  
  
    netdev_device { $hostname: }  
  
    $vlans = {  
        'Blue'    => { vlan_id => 100, description => "This is a Blue vlan" },  
        'Green'  => { vlan_id => 101, description => "This is a Green vLAN" },  
        'Purple' => { vlan_id => 102, description => "This is a Puple vlan" },  
        'Red'    => { vlan_id => 103, description => "This is a Red vlan" },  
        'Yellow' => { vlan_id => 104, description => "This is a Yellow vlan" }  
    }  
  
    create_resources( netdev_vlan, $vlans )  
  
    ...  
}
```

Konfiguration VLAN – Junos II

```
$access_ports = [  
  'ge-0/0/0',  
  'ge-0/0/1',  
  'ge-0/0/2'  
]  
  
$uplink_ports = [  
  'xe-0/0/0',  
  'xe-0/0/2'  
]  
  
netdev_l2_interface { $access_ports:  
  untagged_vlan => Blue  
}  
  
netdev_l2_interface { $uplink_ports:  
  tagged_vlans => keys( $vlans )  
}  
}
```

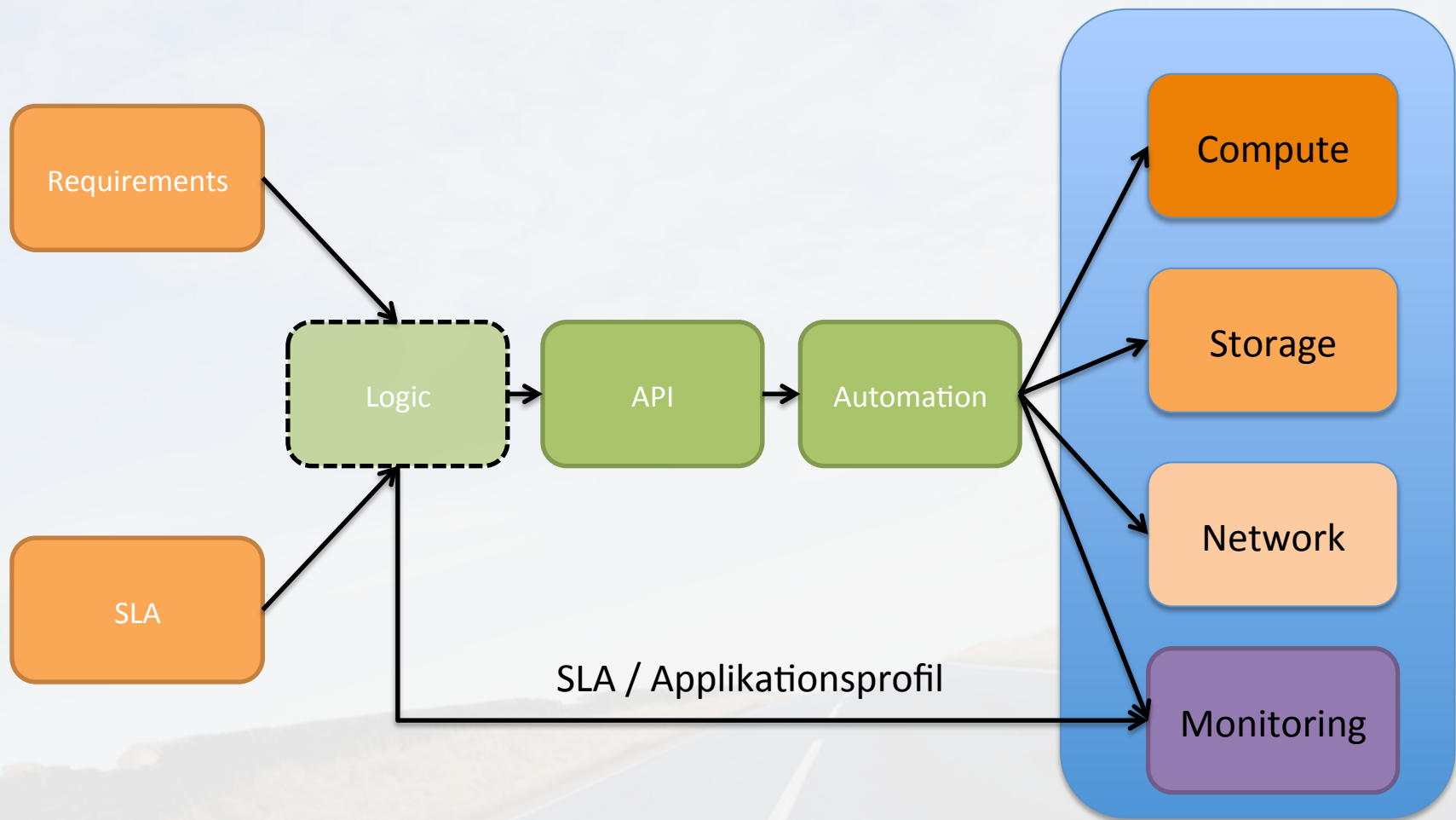
Konfiguration NetApp – Puppet

```
vserver { 'test2':  
    ensure => "present",  
    cmgmt => "172.16.1.83",  
    cuser => "admin",  
    cpass => "netapp123",  
    root_volume => "rootvol", #immutable  
    root_volume_aggregate => "aggr1", #immutable  
    root_volume_security_style => "unix", #immutable  
    aggr_list => "aggr1", #missing create/modify? support  
    language => "C", #immutable  
    snapshot_policy => "default",  
    comment => "test1",  
    antivirus_on_access_policy => "default",  
    quota_policy => "default",  
    name_server_switch => "file",  
}
```



MONITORING

Monitoring Logic



Konfiguration Nagios/Icinga – Chef I

```
% cat roles/monitoring.rb
name "monitoring"
description "Monitoring server"
run_list(
  "recipe[nagios::server]"
)

default_attributes(
  "nagios" => {
    "server_auth_method" => "htauth"
  }
)

% knife role from file monitoring.rb
```

Konfiguration Nagios/Icinga – Chef II

1. Searches for users in 'users' databag belonging to to 'sysadmin' group and authorizes them to access the Nagios web UI and receive notification e-mails.
2. Searches all available roles/environments and builds a list which will become the Nagios hostgroups.
3. Places nodes in Nagios hostgroups by role / environment membership.
4. Installs various packages required for the server.
5. Sets up configuration directories.
6. Moves the package-installed Nagios configuration to a 'dist' directory.
7. Disables the 000-default VirtualHost present on Debian/Ubuntu Apache2 package installations.
8. Templates configuration files for services, contacts, contact groups, templates, hostgroups and hosts.
9. Enables the Nagios web UI.
10. Starts the Nagios server service

Konfiguration Nagios/Icinga – Chef III

.....

node['nagios']['home'] - Nagios main home directory, default "/usr/lib/nagios3"

node['nagios']['conf_dir'] - location where main Nagios config lives, default "/etc/nagios3"

node['nagios']['config_dir'] - location where included configuration files live, default "/etc/nagios3/conf.d"

node['nagios']['log_dir'] - location of Nagios logs, default "/var/log/nagios3"

node['nagios']['cache_dir'] - location of cached data, default "/var/cache/nagios3"

node['nagios']['state_dir'] - Nagios runtime state information, default "/var/lib/nagios3"

node['nagios']['run_dir'] - where pidfiles are stored, default "/var/run/nagios3"

.....

Konfiguration Nagios/Icinga – Puppet I

Nagios Module

```
class nagios::export {
  @@nagios_host { $::fqdn:
    address      => $::ipaddress,
    check_command => 'check-host-alive!3000.0,80%!5000.0,100%!10',
    hostgroups   => 'all-servers',
    target       => "/etc/nagios/resource.d/host_${::fqdn}.cfg"
  }
}
```

site.pp

```
node default {
  include nagios::export
}
```

Konfiguration Nagios/Icinga – Puppet II

init.pp - Apache

```
@@nagios_service { "check_http_${hostname}":  
    use => "check-http",  
    host_name => "$fqdn",  
}
```

```
@@nagios_service { "check_http_processes_${hostname}":  
    use => "remote-nrpe-httpd-procs",  
    host_name => "$fqdn",  
}
```

init.pp - SSH

```
@@nagios_service { "check_ssh_${hostname}":  
    use => "check-ssh",  
    host_name => "$fqdn",  
}
```



ZUSAMMENFASSUNG

Zusammenfassung

- Automatisierte Überwachung mit Applikationsbezug
 - ▶ Qualität und Quantität
- Automatisierung der Installation und Konfiguration
- Es kann auch gern ein Shellsript sein
 - ▶ CMDB-Export oder Excel-Liste
- Keine Überwachung ohne Priorisierung
 - ▶ 8 x 5
 - ▶ 24 x 7
- Integration in Applikationslebenszyklus



FRAGEN & ANTWORTEN



**Fragen
&
Antworten**

NETWAYS GmbH

**Deutschherrnstrasse 15-19
90429 Nürnberg**

Tel: +49 911 92885-0

Fax: +49 911 92885-77

Email: info@netways.de

Twitter: twitter.com/netways

Blog: blog.netways.de