



Python VS Perl

Storage Monitoring per API statt SNMP

Björn Müller
Marcel Denia
comNET GmbH
13.09.2017

Agenda

- Über uns
- Ausgangssituation
- Umsetzung Python
- Umsetzung Perl
- ???

Über uns

Über uns / Ausgangssituation / Python / Perl / ??? / Profit



comNET ist ein Spezialist für IP-Kommunikation und konvergente Netzwerklösungen und garantiert qualifizierte Arbeit im Bereich LAN, Wireless, WAN, Voice- und Video over IP, Security und Unified-Computing.

Wer sind wir?



Björn Müller
IT Business Consultant



Marcel Denia
Ausbildung
Fachinformatiker für Systemintegration

Team haus12 Open Source

Planung, Entwicklung und Betreuung von Open-Source Lösungen

Nagios®



Die Ausgangssituation

Über uns / **Ausgangssituation** / Python / Perl / ??? / Profit

Die Ausgangssituation



Huawei OceanStor 5500 V3

Storage-Appliance

SAN/NAS

Verwaltet in maximaler Ausbaustufe 15 PB

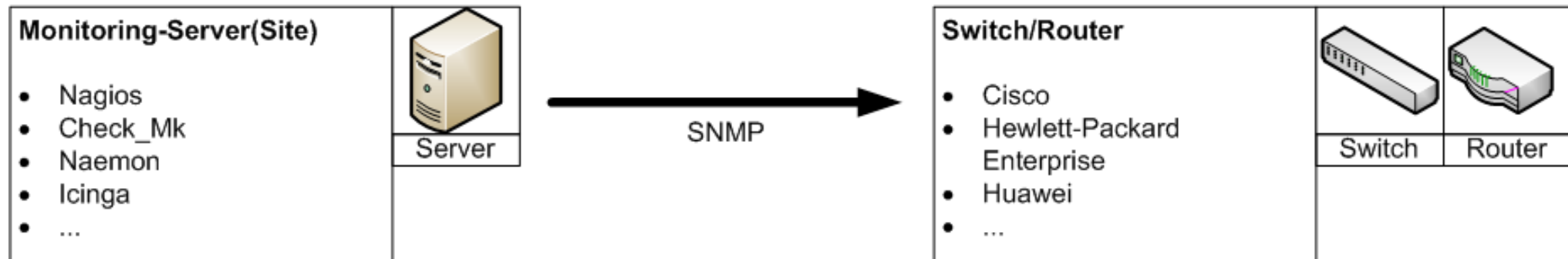
Integration in VMWare ESX, Microsoft Hyper-V und Citrix

Wie funktioniert Monitoring?

1. Monitoring per Agent



2. Monitoring via SNMP



SNMP in der Praxis

- Netzwerk-Infrastruktur benutzt SNMP
- Andere wenden sich von SNMP ab
 - VMWare ESX
 - innovaphone
 - Huawei OceanStor

Gründe

- Vergleichsweise komplexer Standard
 - ⇒ Inkompatible Implementierungen
- Modellierung schwierig
- Schwer menschenlesbar
- Traps nicht zuverlässig

```
h3cEntityExtTemperature OBJECT-TYPE
    SYNTAX INTEGER
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "..."
```

```
::= {
    h3cEntityExtStateEntry 12
}
```

⇒ 1.3.6.1.4.1.25506.2.6.1.1.1.1.12

Und andere meckern über IPv6...

SNMP is dead, all hail REST

- **R**epresentational **S**tate **T**ransfer
- Konzept zur Gestaltung von HTTP-/Web-APIs
- Sehr verbreitet im Web
- Üblicherweise JSON zur Serialisierung

```
$ wget http://example.com/api/services?name=ssh -o /dev/null -O -  
  
{  
    "name": "ssh",  
    "port": 22,  
    "privileged": 1  
}
```

Neues Protokoll, neue Herausforderungen

- Authentifizierung
 - Aufgabe der Anwendung
 - Nicht standardisiert
 - => Gefühlt 30000 Möglichkeiten, sich zu Authentifizieren
- Repräsentation
 - JSON
 - YAML
 - XML(?!)
- Aufbau der URIs
- Wie verhalten sich HTTP-Methoden
- Interpretation der gelieferten Daten



File Edit View Go Bookmarks Tools Settings Help



Previous



Next

Fit Page



Zoom Out



Zoom In



Browse



Zoom



Selection



Contents



Thumbnails



Reviews



Bookmarks



**OceanStor 5300 V3&5500 V3&5600 V3&5800
V3&6800 V3 Storage System V300R003C20 REST
Interface Reference**

Issue 01
Date 2016-11-10

HUAWEI TECHNOLOGIES CO., LTD.



1

1

of 1504



13

Umsetzung Python

Über uns / Ausgangssituation / **Python** / Perl / ??? / Profit

Ansatz Python / Library

- Warum Python?
- Warum Bibliothek?
- Ausgliederung in Bibliothek:
 - Authentifizierung
 - Kommunikation
 - GET
 - POST
 - PUT
 - DELETE



Ansatz Python / Check

- Argument Parser
- QueryItems
- Path
- Alarm – Status Mapping
- Eigentlicher Check

```
from huawei import oceanstor

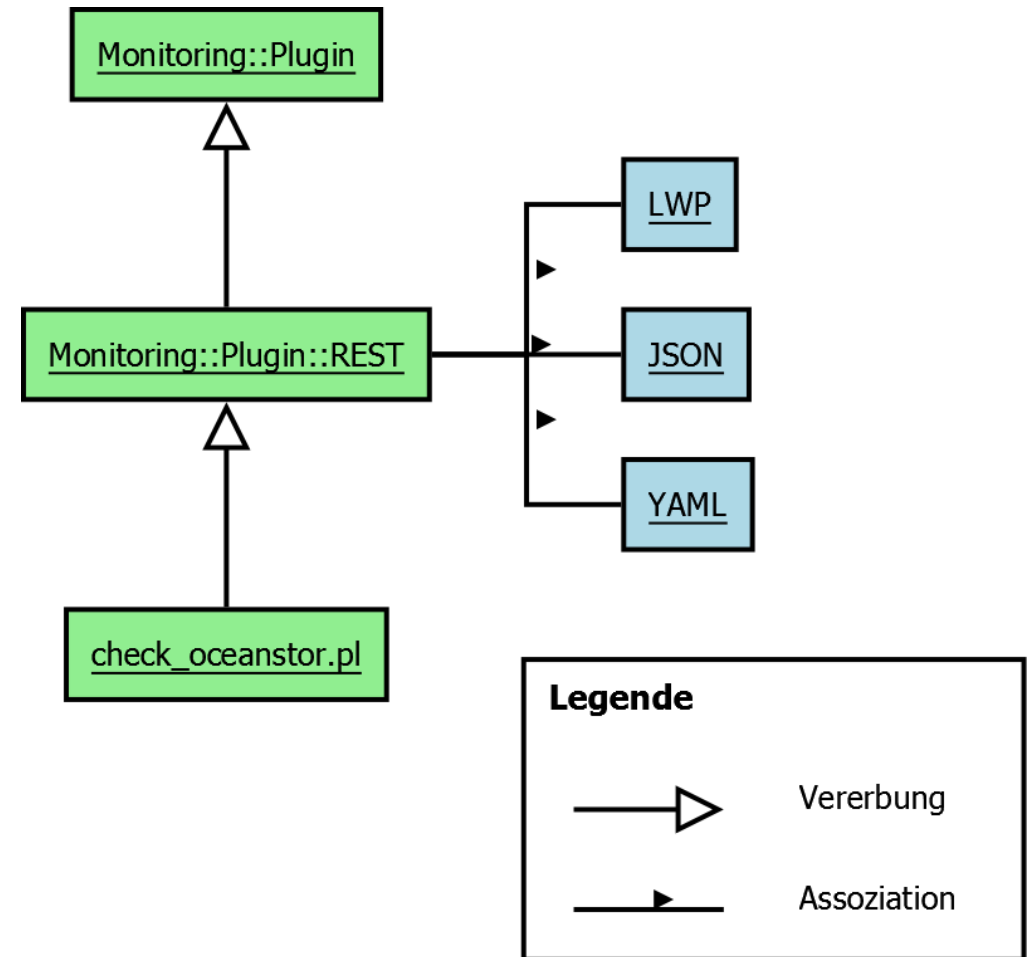
path = 'alarm/currentalarm'
query_items = [
    'sequence',
    'description',
    'level',
    'location',
    'startTime',
]
```

Umsetzung Perl

Über uns / Ausgangssituation / Python / **Perl** / ??? / Profit

Konzept Perl

- Auf Monitoring::Plugin aufbauen
- Eigenständiges Modul zum REST-Handling
 - Authentifizierung
 - JSON/YAML/...
 - Nicht an OceanStor gebunden
- Darauf aufbauender Check



Monitoring::Plugin::REST

- Monitoring::Plugin oder Nagios::Plugin
 - Perl >=5.10.0
 - LWP
- Konfigurierbare Authentifizierung
- Transport per JSON und YAML

check_oceanstor.pl

- Monitoring::Plugin::REST
- Huawei OceanStor 5300 V3/5500 V3/5600 V3/5800 V3/6800 V3
- Interpretation und Bewertung der gelieferten Daten

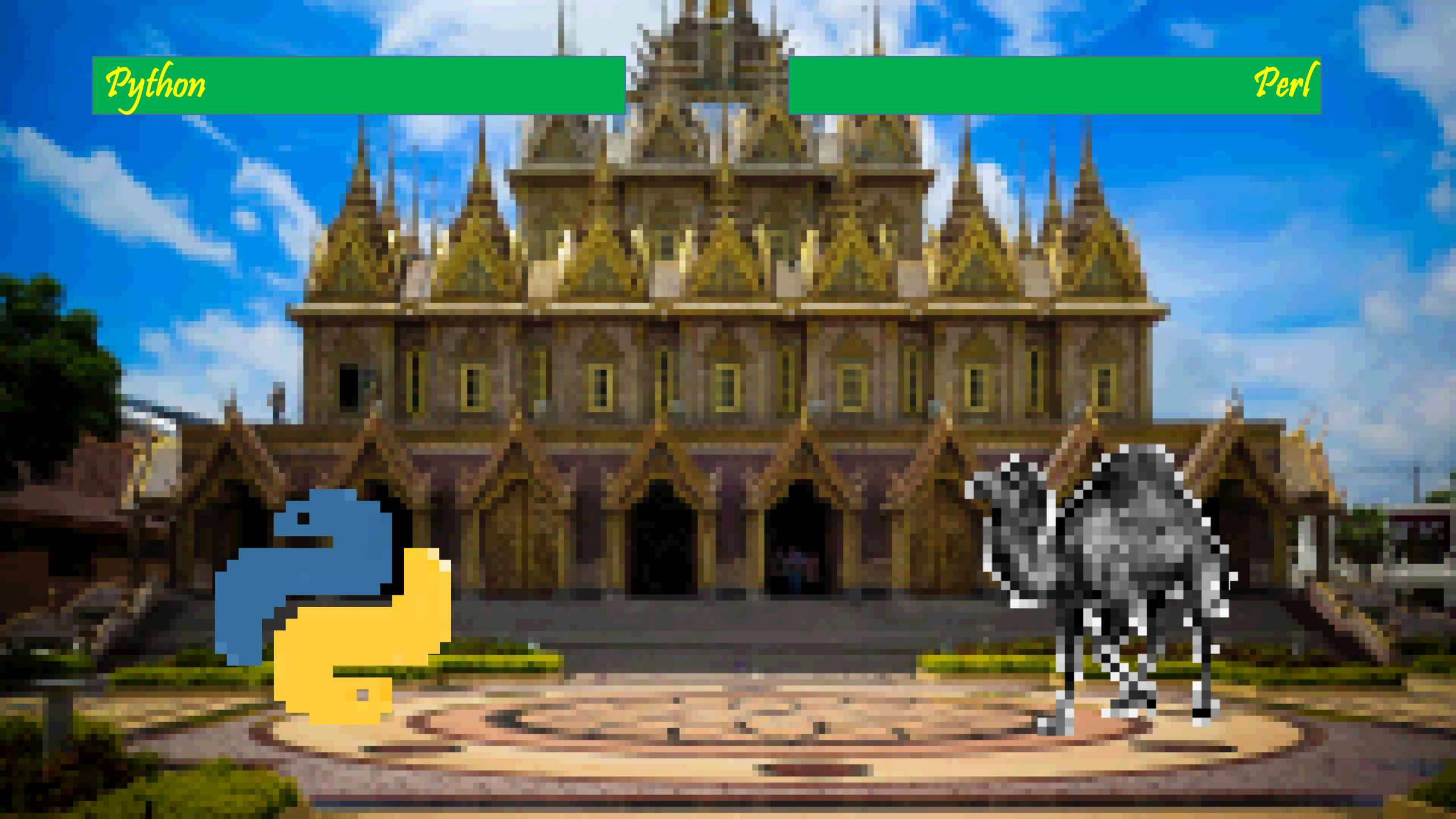
???

Über uns / Ausgangssituation / Python / Perl / ??? / Profit



Python

Perl



Python

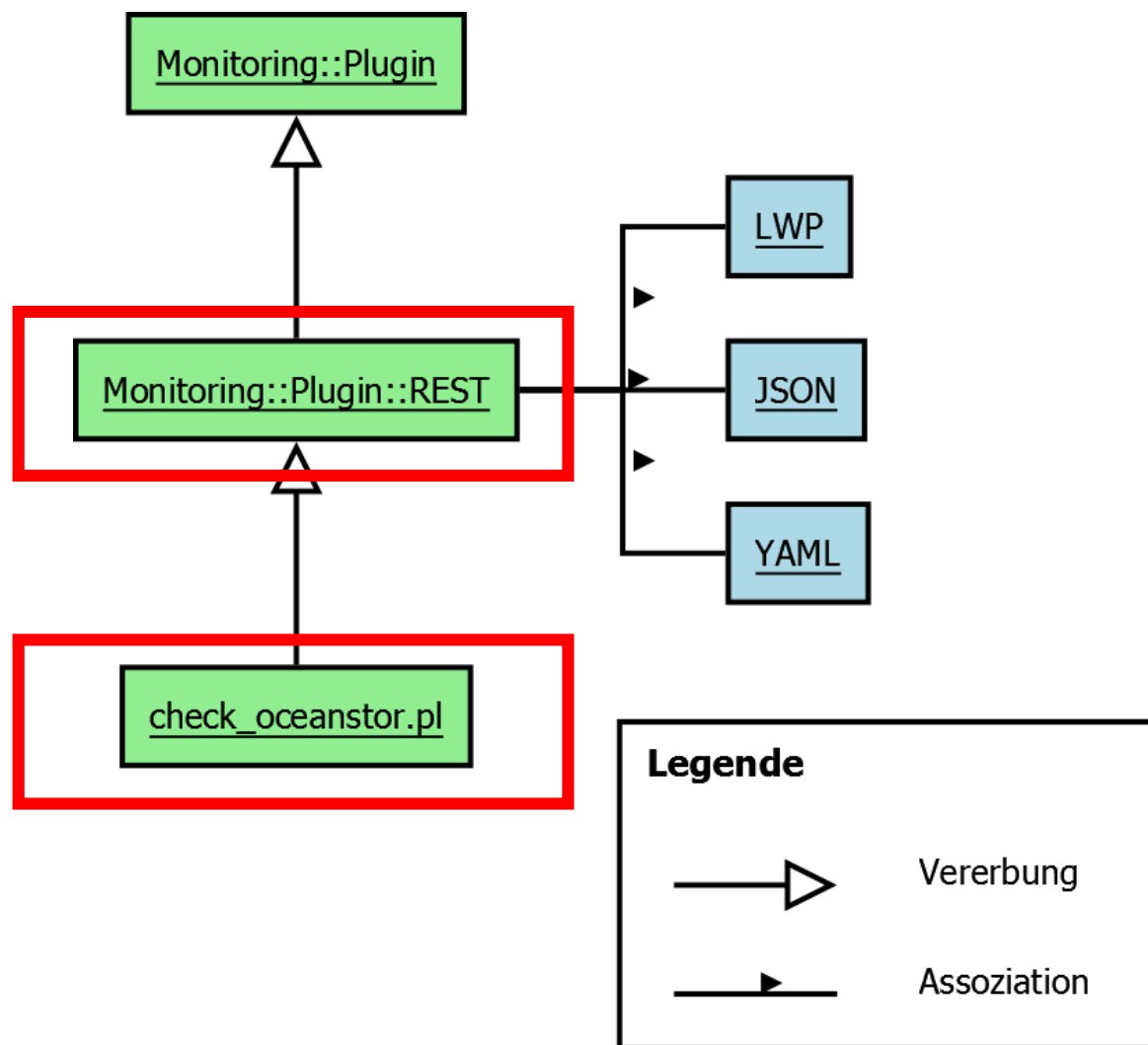
Perl

Platz	Programmiersprache
1	Java
2	C
3	C++
4	C#
5	Python
6	PHP
7	JavaScript
8	Visual Basic .NET
9	Perl
10	Ruby

Tiobe Index[2] September 2017

Python

Perl



Python

Perl

```
root@themachine:~#
```





Python

Perl

```
#!/usr/bin/perl

use strict;
use warnings;
use Monitoring::Plugin::REST;

my $plugin = Monitoring::Plugin::REST->new(
    host => 10.10.10.10,
    user => 'testuser',
    password => 'sicher',
    authentication => [
        { method => 'get', endpoint => '/login', keep_cookies => 1 },
    ],
);

my $status = $plugin->fetch('/port_status');

# Analyze $status, exit accordingly
```

Python

Perl

Nagios®

&



Mathias Kettner

Python

Perl

```
marcel@dell:~/libyaml-perl-1.21$ make test
PERL_DL_NONLAZY=1 PERL_USE_UNSAFE_INC=1 "/usr/bin/perl" "-
MExtUtils::Command::MM" "-MTest::Harness" "-e" "undef
*Test::Harness::Switches; test_harness(0, 'blib/lib', 'blib/arch')"
t/*.t
t/000-compile-modules.t ... ok
t/2-scalars.t ..... ok
t/basic-tests.t ..... ok
t/bugs-emailed.t ..... ok
t/bugs-rt.t ..... ok
...
marcel@dell:~/libyaml-perl-1.21$
```

Software Failure. Press left mouse button to continue.

Guru Meditation #0100000C.000FE800



Profit

Über uns / Ausgangssituation / Python / Perl / Mortal comNET / **Profit**

Die bessere Lösung

Es kommt halt ganz darauf an:

Die Python-Lösung ist **spezieller zugeschnitten**

Die Perl-Lösung ist **flexibler**

Das richtige Werkzeug für die Aufgabe wählen!

Zukunft

- Laufendes Session-Cookie aufbewahren
- Simulations-Möglichkeit wie snmpsim oder check_nwc_health
- Mehr Checks!

Quellen

[1] Huawei API reference

[2] TIOBE programming language index

- <https://www.tiobe.com/tiobe-index/>

Links

comNET GmbH

<https://www.comnetgmbh.com>

GitHub

<https://github.com/comnetgmbh>



Björn Müller
bjoern.mueller@comnetgmbh.com



Marcel Denia
marcel.denia@comnetgmbh.com

