SysDB – System DataBase

Alle Systeminformationen auf einen Blick

Sebastian 'tokkee' Harl
<tokkee@sysdb.io>

FrOSCon
Free and Open Source Software Conference

22. August 2015

Disclaimer

HINWEIS:
SysDB ist noch in der Entwicklung
und wird noch nicht als stabil angesehen[1].

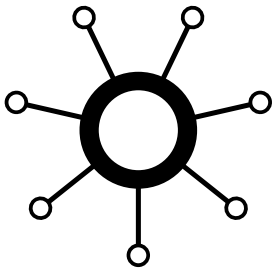
Flaming, Bashing oder andere Formen von konstruktivem
Feedback sind sehr willkommen ... oder **Mitwirkung** :-)

[1] I.S.v. Änderungen an Schnittstellen

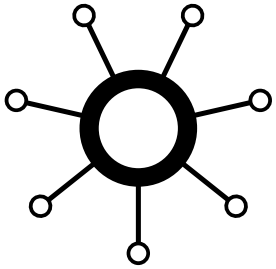
Motivation / Hintergrund

Motivation

Motivation



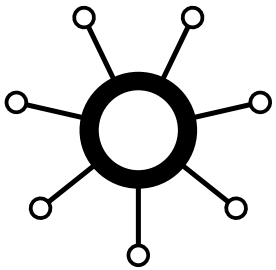
Motivation



Motivation

Naemon

ICINGA

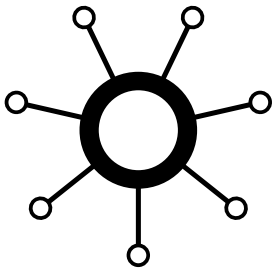


- Database Server Foo
- External Gateway
- Zooperserver

Motivation

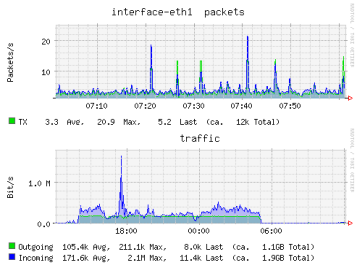
Naemon

ICINGA



- Database Server Foo
- External Gateway
- Zooperserver

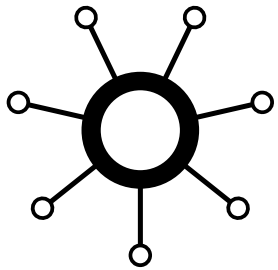
gw1.domain.tld



Motivation

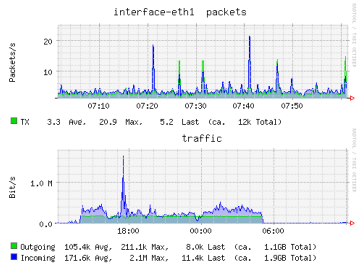
Naemon

ICINGA



- Database Server Foo
- External Gateway
- Zooperserver

gw1.domain.tld



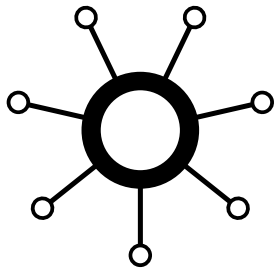
gw1.domain.tld

```
arch=amd64
lsbdistid=Debian
loc=BayArea
```

Motivation

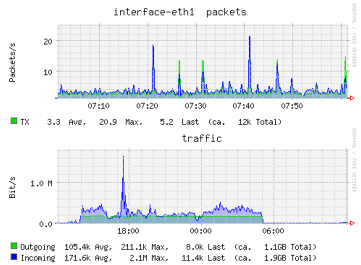
Naemon

ICINGA



- Database Server Foo
- External Gateway
- Zooperserver

gw1.domain.tld

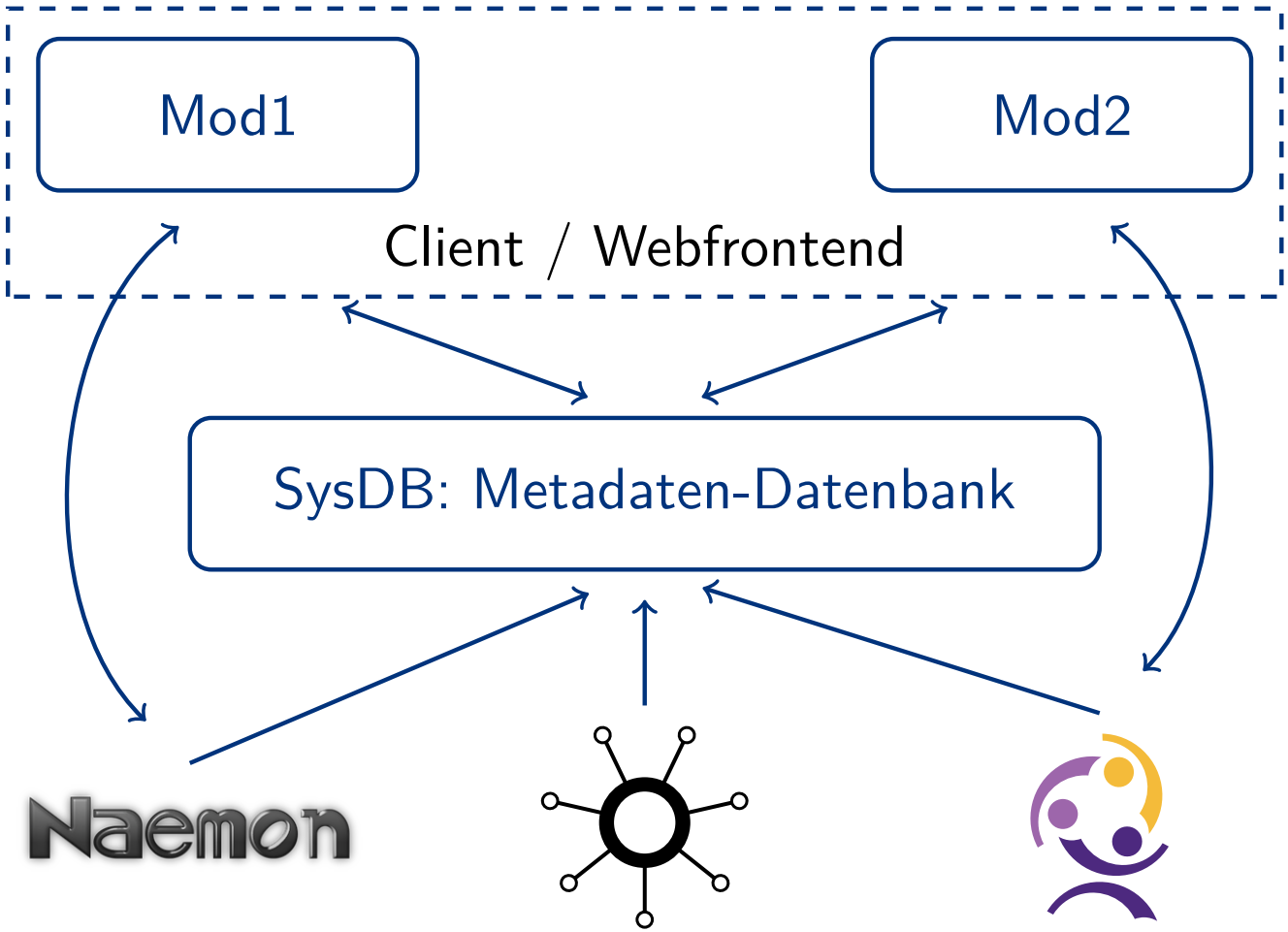


gw1.domain.tld

```
arch=amd64
lsbdistid=Debian
loc=BayArea
```

Die System DataBase

SysDB – Platform für System-Informationen



SysDB – Überblick

- SysDB aggregiert Informationen zu beliebigen Hardware- und Software-Systemen zu einem Gesamtüberblick
- Fokus auf Metadaten und Verknüpfung zu anderen Informationen
- Einfache Beispiele
  - Hosts und deren Eigenschaften („facts“)
  - Services und deren Eigenschaften
  - Metriken (Verweis auf Performance-Daten)
  - Monitoring Informationen
- SysDB sammelt diese Informationen von verschiedenen Systemen und korreliert zusammengehörige Informationen
- (Einfache) Web-Oberfläche

SysDB – wichtige Eigenschaften

- `https://sysdb.io/`, `https://github.com/sysdb`
  - CI: `https://travis-ci.org/sysdb/sysdb`
  - $\sim$ 80% Code (Funktion) Unit-Test Coverage im Core
- BSD Lizenz
- Geschrieben in C und Go (WebUI)
- Generische Speicherung von aggregierten Systeminformationen
- Multi-threaded, event-based client/server Architektur
- Einfach erweiterbar (Plugin API)
- Einfaches Netzwerk-Protokoll und Abfragesprache

SysDB – Plugins

Aktuell verfügbare Backends (Kollektoren):

- `collectd::unixsock` – Abfrage von collectd (unixsock Plugin)
  → Host Metriken
- `mk-livestatus` – Abfrage von Monitoring Systemen (Nagios, Naemon, Icinga, Shinken) mittels Check_MK Livestatus
  → Hosts und Services (Metriken geplant)
- `facter` – Abfrage von lokalen facter „facts“
  → Host Attribute
- `puppet::store-configs` – Abfrage von Puppet
  → Host Attribute

SysDB – Plugins (2)

Store Plugins:

- `store::memory` – In-Memory Datenbank
- `store::network` – Versand (Replikation) von gesammelten Daten an eine entfernte Instanz

Time-series Plugins:

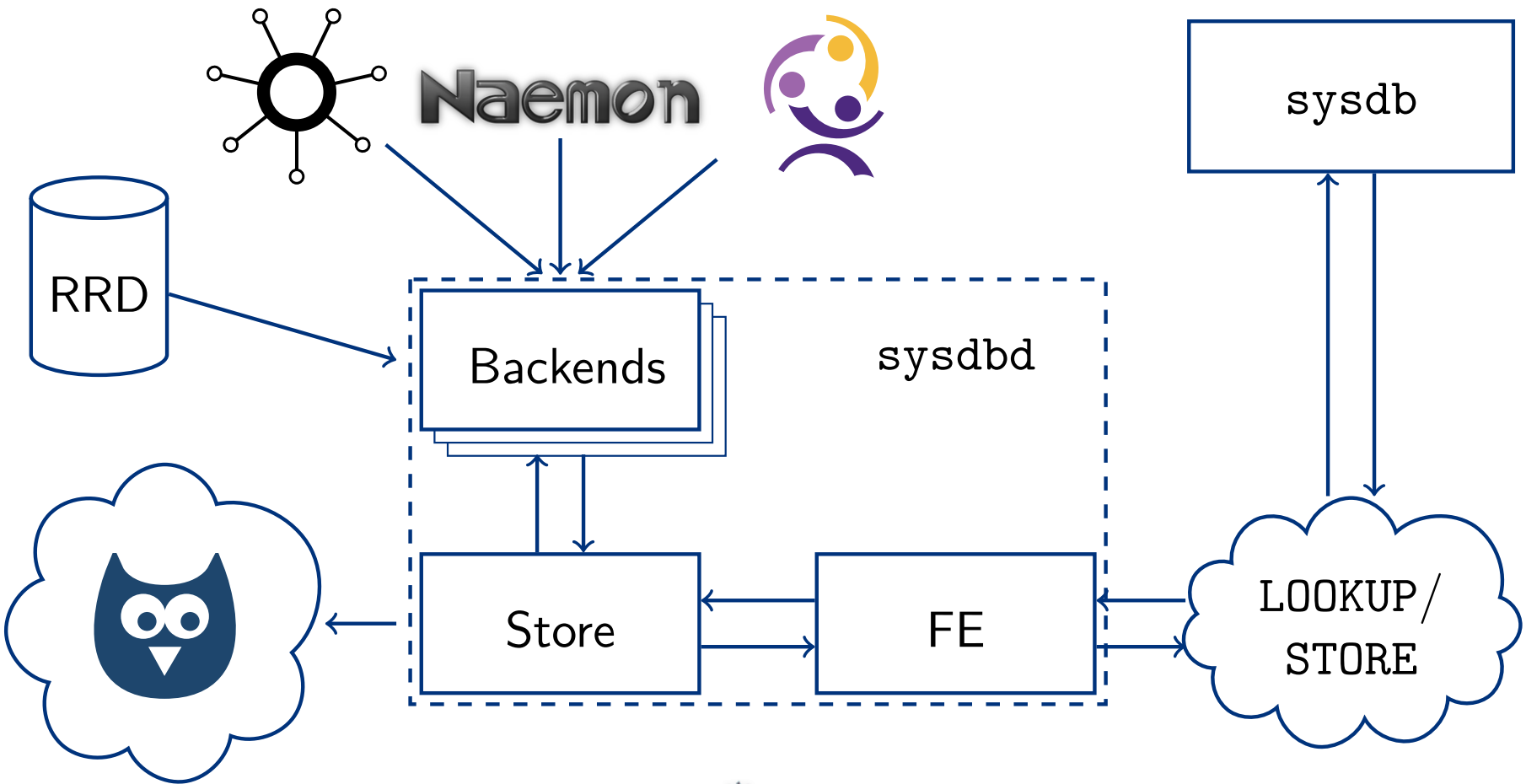
- `timeseries::rrdtool` – Abfrage von Timeseris aus lokalen RRD Dateien

SysDB – Plugins (3)

Verschiedene Plugins:

- `cname::dns` – Normalisierung von Hostnamen mittels DNS
- `syslog` – syslog Logging

SysDB – Architektur



Der SysDB Store

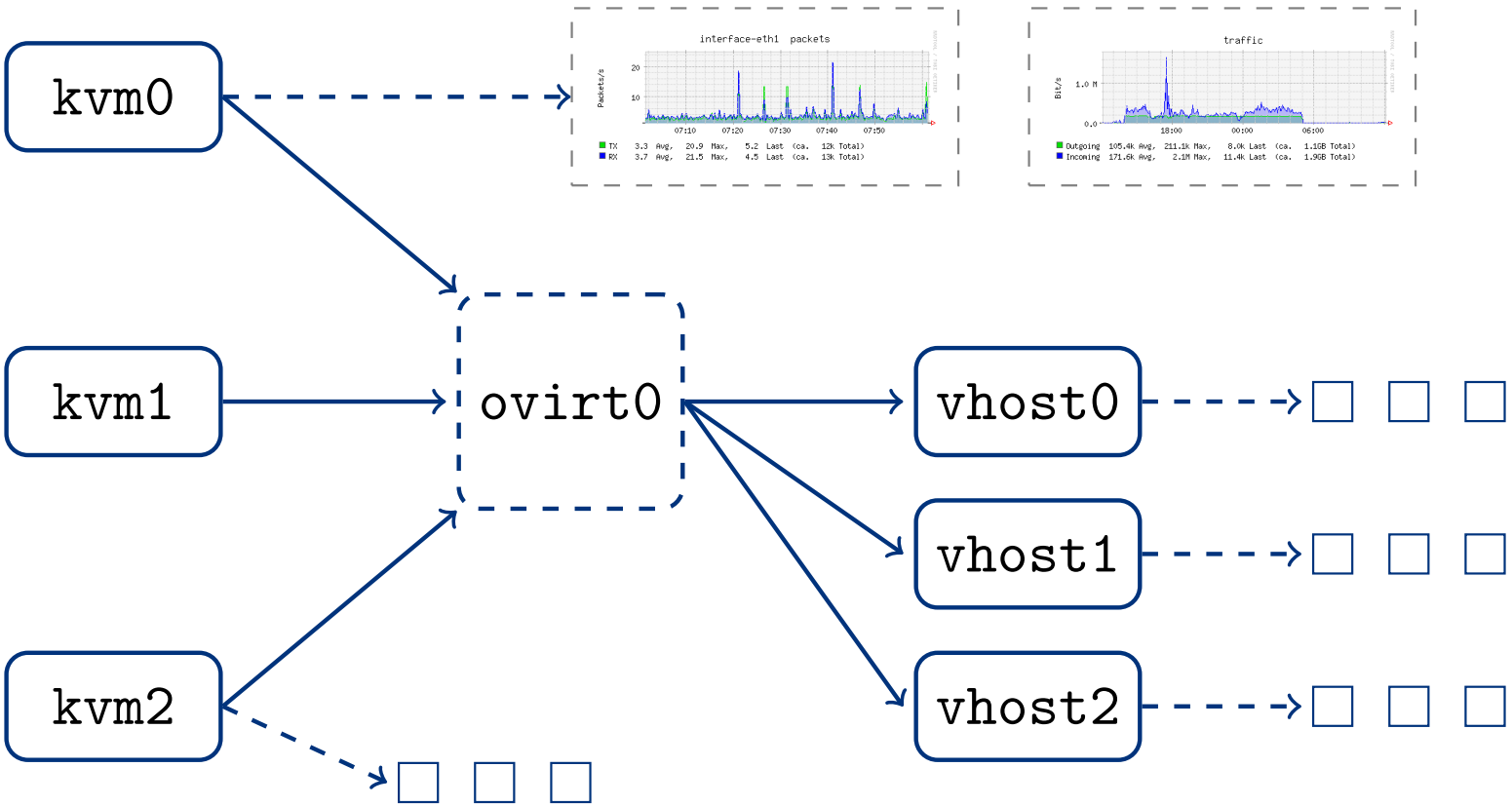
- Speichert generische Objekte (via Plugins)
- Normalisierung von Hostnamen
- Jedes Objekt speichert Zeitpunkt der letzten Aktualisierung und automatisch berechnetes Update-Intervall

- Schnittstelle zur Datenabfrage
- Generischer Zugriff auf Timeseries Daten
- JSON als externe Repräsentation von Objekten

Der SysDB Store – Objekte

- Host – jede Art physikalischer Ressource
- Service – Dienste (im weitesten Sinne)
- Metric – Bezeichner für Metriken
  - → Timeseries Daten werden **nicht** in SysDB gespeichert
- Attribute – Eigenschaften von Hosts und Services
  - → verschiedene Datentypen unterstützt
- Attribute aller Objekte:
  - eindeutiger Name
  - Zeitpunkt des letzten Updates
  - Update Intervall
  - Liste von Backends

Der SysDB Store – Beispiel



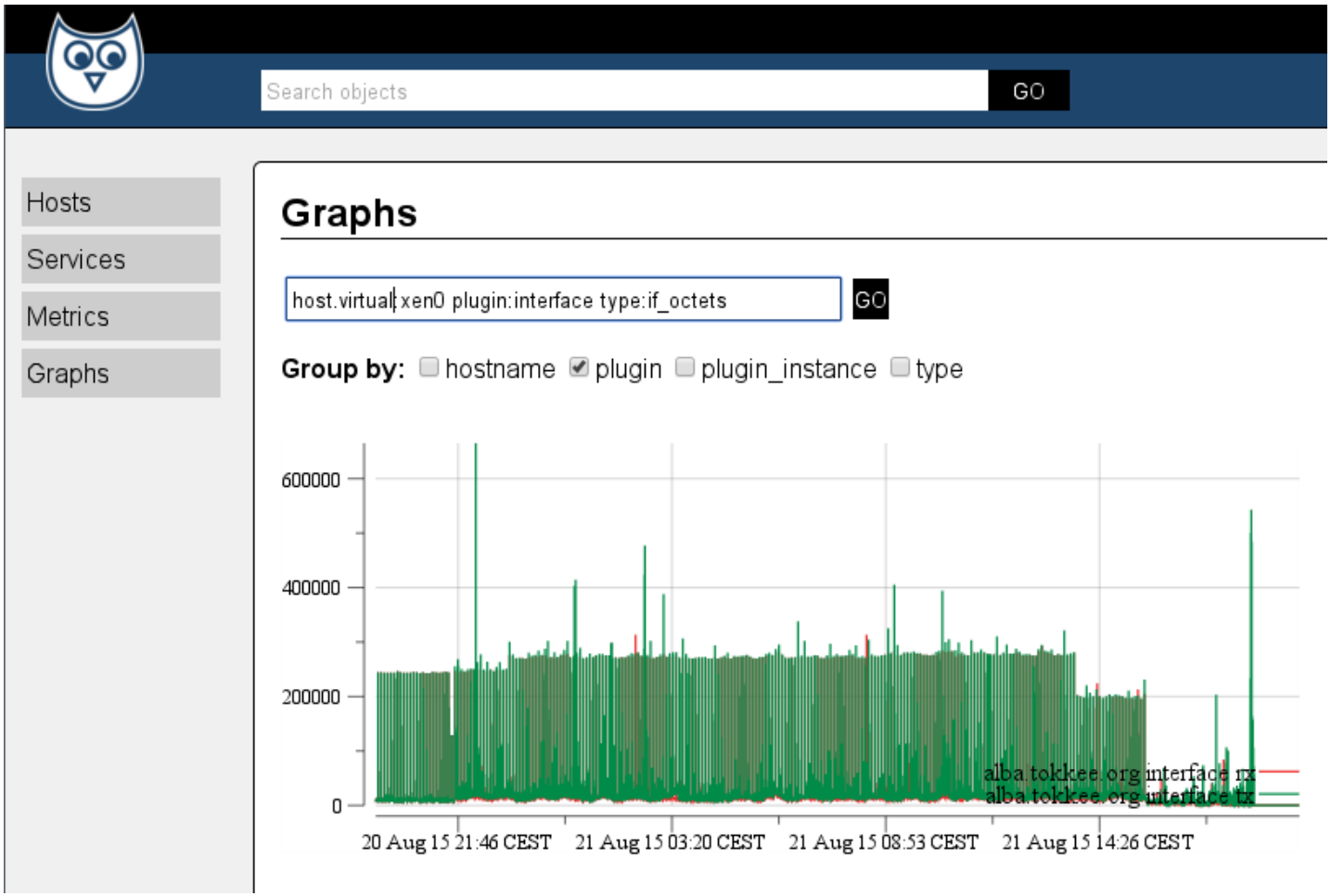
Der SysDB Client

- Interaktiver Client für SysDB
- Verbindet sich mit einem SysDB Daemon
- Interaktive Kommando-Shell
- Zeigt interaktive Log-Meldungen vom Server an

Die SysDB WebUI

- Datenbank Browser
- Suche mittels einfacher Abfragesyntax
- Metrik-Graphen
- Dynamische Graphen mit Aggregierung / Gruppierung

Die SysDB WebUI



SysQL – Die SysDB Query Language

- Entfernt an SQL angelehnt
- Aber es ist nicht SQL[1]
- Auflistung und Abfrage von Hosts, Services, Metriken
- Abfrage von Timeseries Daten (von einem Backend Data-Store)
- Speichern und Aktualisierung von Objekten

[1] Eine DSL war die einfachste Option ;-)

SysQL – Die SysDB Query Language (2)

- `LIST hosts|services|metrics`
- `FETCH host <hostname>`
- `FETCH service|metric <hostname>.<name>`
- `LOOKUP hosts|services|metrics MATCHING <condition>`
- `... FILTER <condition>`
- `TIMESERIES <host>.<metric>`
  `START 2015-01-01 END 2015-12-31`
- `STORE <type> ...`

→ `https://sysdb.io/manpages/head/sysdbql.7.html`

SysQL – Beispiel

```
sysdb=> LIST hosts FILTER age < 2 * interval;
[{
    "name": "monitor.lxc.tokkee.net",
    "last_update": "2015-04-03 10:26:41 +0200",
    "update_interval": "5m4s",
},{
    "name": "puppet.lxc.tokkee.net",
    "last_update": "2015-04-05 11:04:08 +0200",
    "update_interval": "5m2s"
}]
```


SysQL – Beispiel (2)

```
sysdb=> LOOKUP hosts MATCHING attribute['arch'] = 'amd64'
                                AND ANY service =~ 'postgres';
{ "name": "db.lxc.tokkee.net",
  "last_update": "...", "update_interval": "10s",
  "attributes": [{
      "name": "architecture", "value": "amd64",
      "last_update": "...", "update_interval": "5m3s"
  },{ ... }],
  "services": [{
    "name": "PostgreSQL Master",
    "last_update": "...", "update_interval": "5m"
  },{ ... }],
  "metrics": [{...}]
}
```


SysQL – Bedingungen

Abfrage- und Filter-Bedingungen:

```
MATCHING attribute['architecture'] = 'amd64'
         AND ANY service =~ 'postgres'

MATCHING name =~ '\.tokkee\.net$'
MATCHING host.attribute['foo'] = 'bar'

FILTER age >= 3 * interval

FILTER NOT ANY backend =~ 'mk-livestatus'
FILTER name IN ['hostA', 'hostB']
```


Anwendungsbeispiele

Anwendungsbeispiele

- Mächtige Web-Frontends für mehrere Backends
  - → Informationen korrelieren oder annotieren
  - → zentrales Dashboard
  - → Integration von spezifischeren Frontends
- Abgleich diverser Backends (Monitoring)
  - → Welche Hosts / Services fehlen in welchem Backend?
  - → Basis für Businessprozesse: Wie ist der Gesamtstatus meiner Windows-Systeme in Rechenzentrum XYZ?
- Erweiterung von CMDBs durch dynamische Informationen

Ausblick: künftige Entwicklungen

Ausblick: künftige Entwicklungen

- Bessere und mehr Integration mit anderen Systemen
- Persistenter Store: RDBMS, Graphdatenbank (?)
- Interface zur Abfrage von anderen Live-Daten (z.B. Monitoring-Status) von Backends
- Verteilte Architektur (HA und Load-Balancing)
- Erweiterbares Web-Interface
- ...
- Send Patches!

SysDB – System DataBase

Danke für die Aufmerksamkeit

Fragen, Kommentare, Rants?

https://sysdb.io/, **https://github.com/sysdb**

https://sysdb.io/+ https://sysdb.io/facebook

@SystemDatabase