SysDB – System DataBase

A system management and inventory collection service

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

Icinga Camp San Francisco 2014
September 25, 2014

Disclaimer

WARNING:
SysDB is still under heavy development
and not considered stable yet[1].

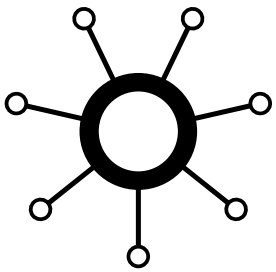
Flaming, bashing or other forms of constructive
feeback are very appreciated ... also, contributions :-)

[1] That is, interfaces may still change quickly.

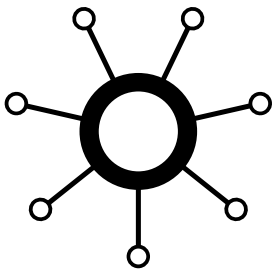
Background / Motivation

Motivation

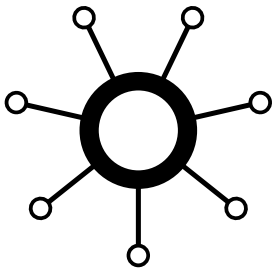
Motivation



Motivation

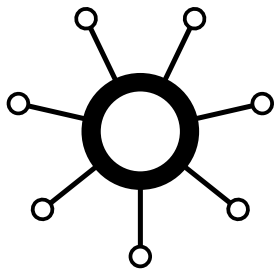


Motivation



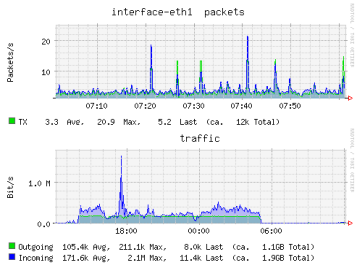
- Database Server Foo
- External Gateway
- Zooperserver

Motivation

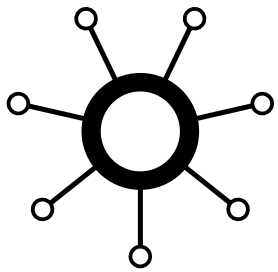


- Database Server Foo
- External Gateway
- Zooperserver

gw1.domain.tld

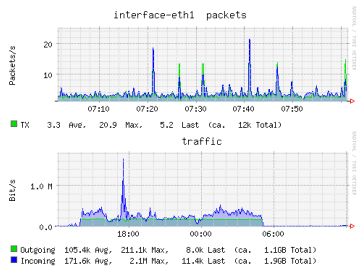


Motivation



- Database Server Foo
- External Gateway
- Zooperserver

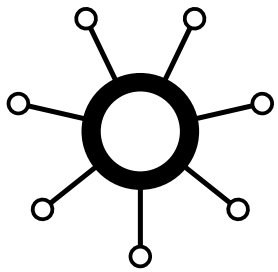
gw1.domain.tld



gw1.domain.tld

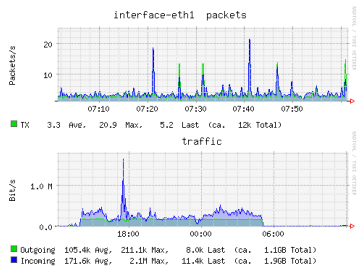
```
arch=amd64
lsbdistid=Debian
loc=BayArea
```

Motivation



- Database Server Foo
- External Gateway
- Zooperserver

gw1.domain.tld

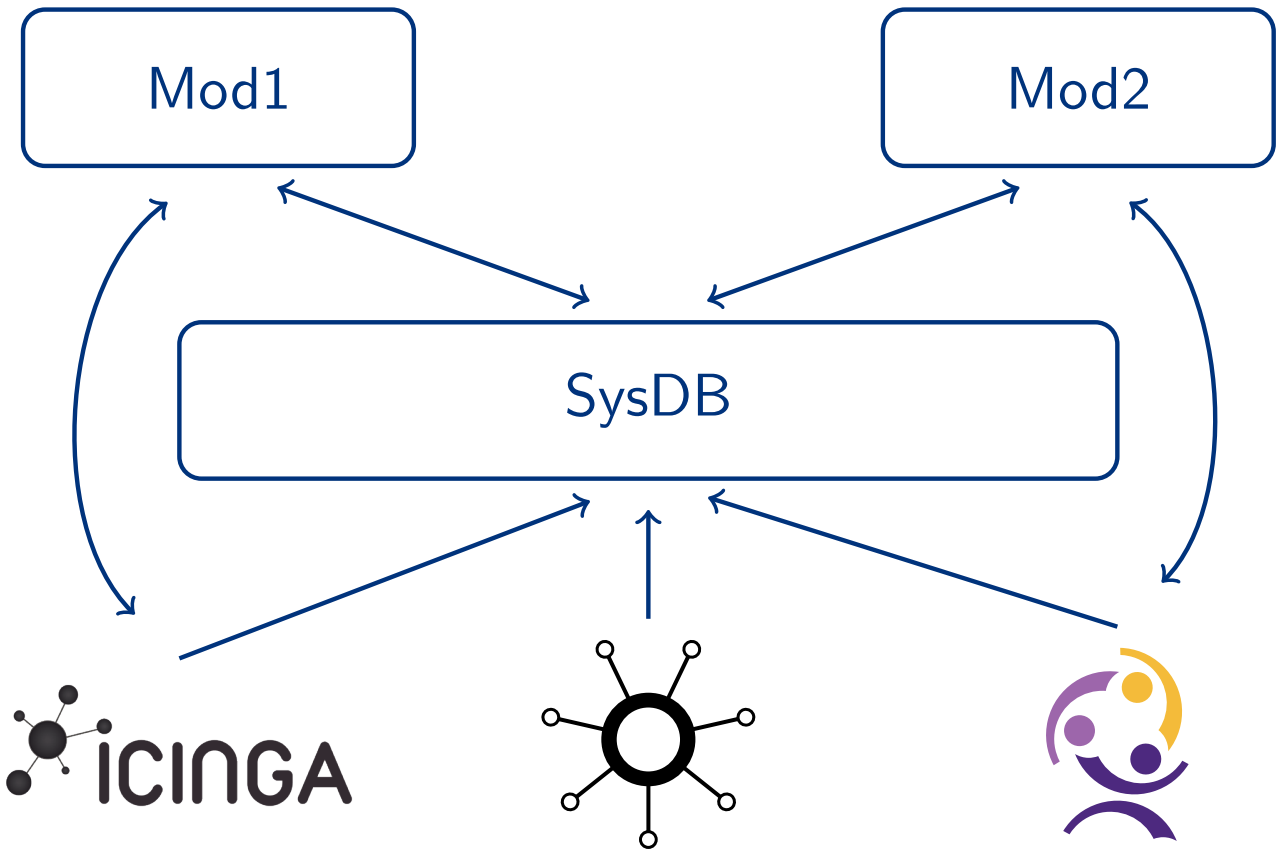


gw1.domain.tld

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

The System DataBase

SysDB – System information platform



SysDB – Overview

- SysDB collects information about arbitrary hardware and software systems for a global overview of the entire infrastructure.
- Focus on meta-data and links to more specific information.
- Simple examples:
  - Hosts and their attributes (“facts”)
  - Services and their attributes
  - Metrics (pointer to performance data)
  - Monitoring information (e.g. current state)
- SysDB collects these information and merges objects from multiple back-ends by hostname.

SysDB – Core features

- `https://sysdb.io/`, `https://github.com/sysdb`
  - CI: `https://travis-ci.org/sysdb/sysdb`
  - $\sim$ 80% code (function) unit-test coverage in the core
- BSD license
- Written in C (UI written in Go WIP)
- Generic store aggregating system information
- Multi-threaded, event-based client/server architecture
- Easy to extend (simple plugin API)
- Simple network protocol and query language
- Most of the code implemented as a library (reusable)

SysDB – Plugins

Currently available collectors (back-ends):

- **`collectd::unixsock`** – query collectd's UNIX socket interface
  → host metrics
- **`mk-livestatus`** – query Monitoring systems (Nagios, Naemon, Icinga, Shinken) using Check_MK Livestatus
  → hosts and services (metrics planned)
- **`puppet::store-configs`** – query Puppet
  → host attributes

Planned: Passive data collection (e.g. using Gearman), Foreman, PuppetDB, $your_favorite_system (send patches!) :-)

SysDB – Plugins (2)

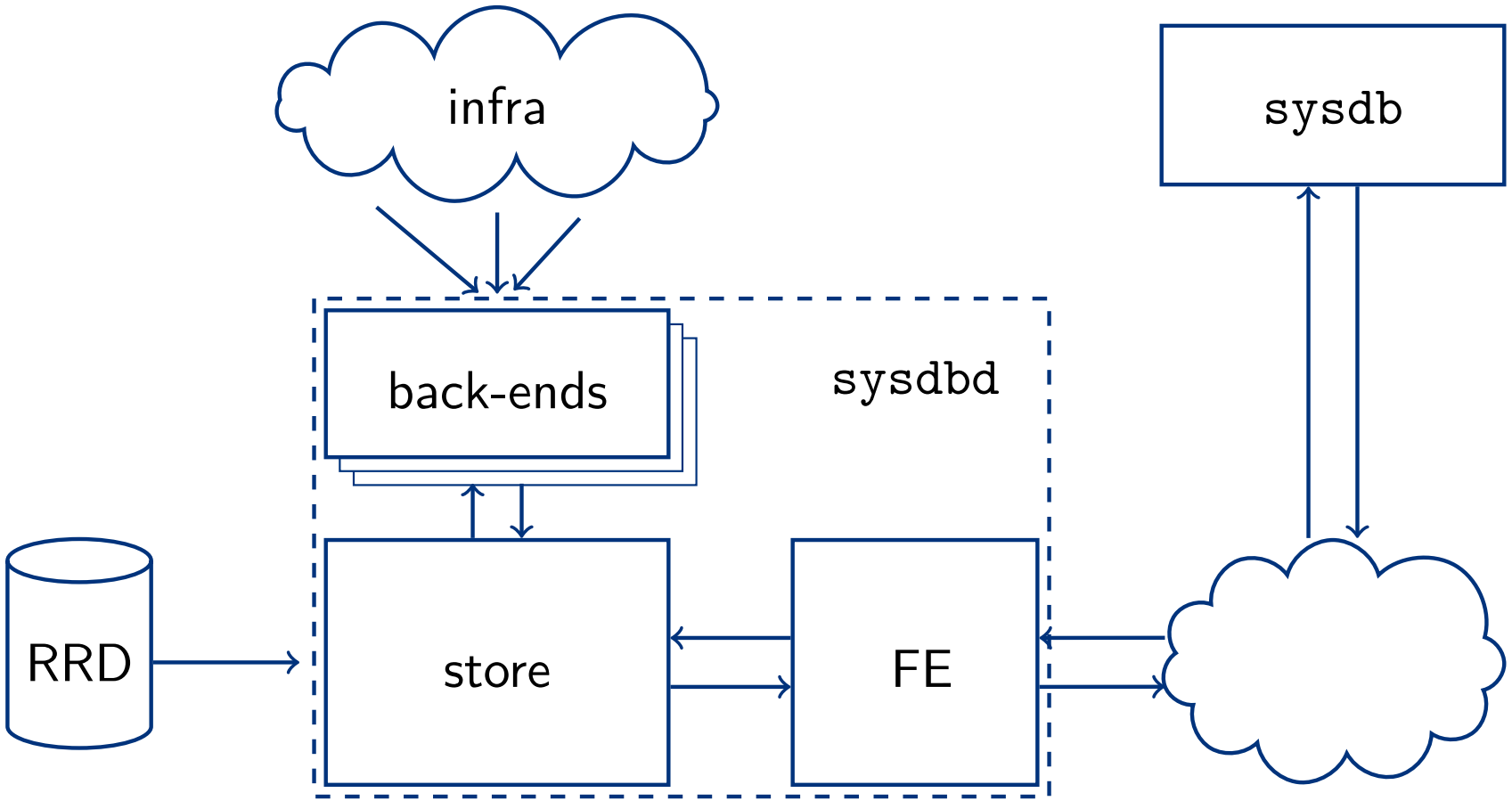
Time-series fetchers:

- `timeseries::rrdtool` – fetch time-series from local RRD files

Misc plugins:

- `cname::dns` – canonicalize host-names using DNS
- `syslog` – syslog logging

SysDB – Architecture



The SysDB Store

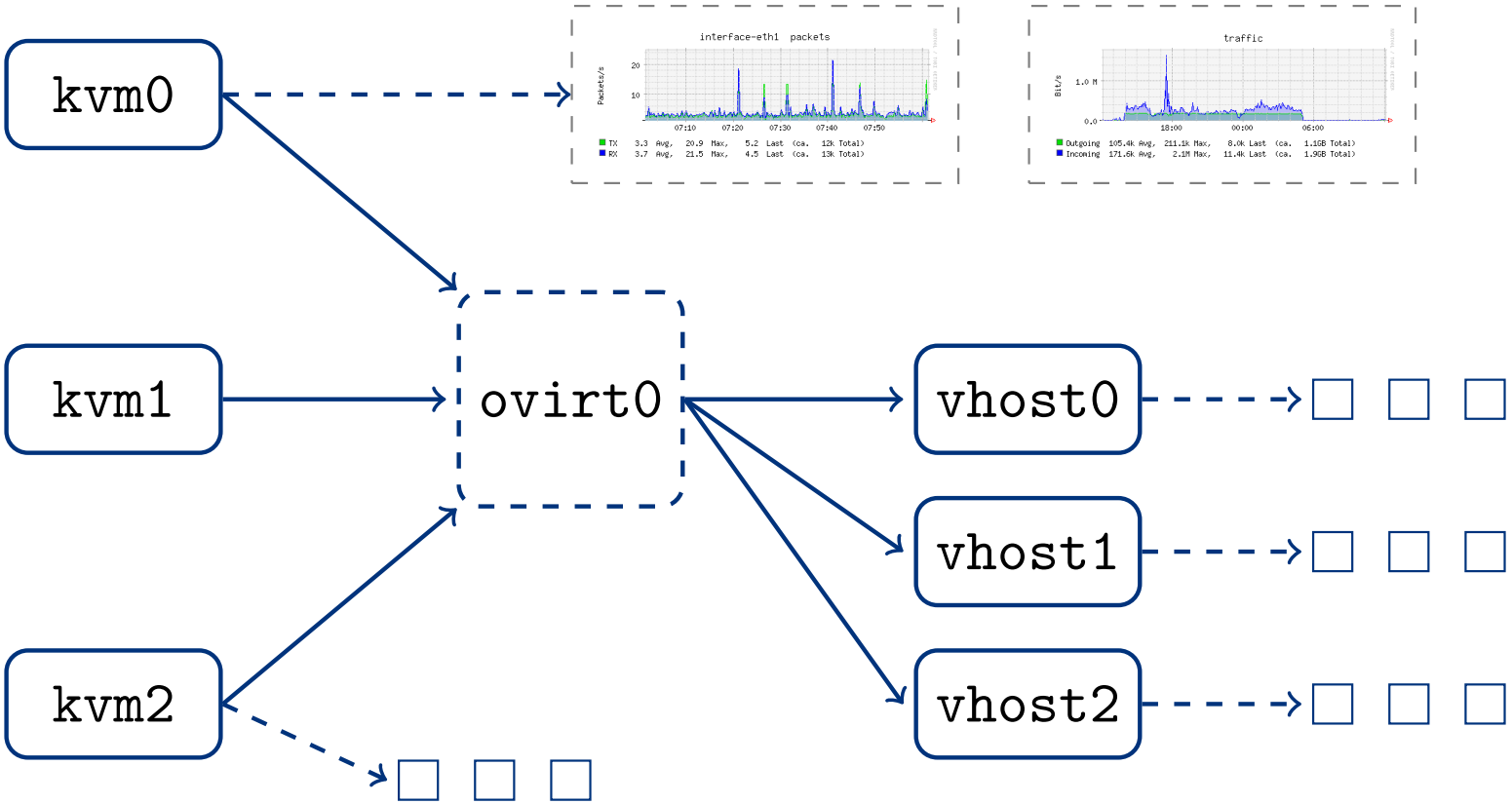
- Stores generic objects (in memory atm)
- Canonicalization of hosts (by their name as provided by back-ends)
- Each object stores the time-stamp of the last update and the (automatically calculated) update interval

- Interface to query data
- Generic access to time-series data
- JSON is the external data representation

The SysDB Store – Stored objects

- Host – any kind of physical resource
- Service – any kind of service
- Metric – identifier for metrics data
  - → time-series data is **not** stored in SysDB
- Attribute – attributes of hosts and services
  - → values support multiple types
- Common core attributes:
  - unique name
  - time-stamp of last update
  - update interval
  - list of back-ends providing the object

The SysDB Store – Example



(Full graph-support not implemented yet.)

The SysDB Client

- Interactive client program for SysDB
- Connects to a SysDB daemon
- Interactive command shell
- Receives and displays asynchronous log messages

The SysDB Query Language

- Remotely similar to SQL
- But it's not SQL[1]
- List and query hosts
- Fetch time-series information (from some back-end data-store)

[1] Yes, a DSL might limit options but it was the easiest for now – believe it or not ;-)

The SysDB Query Language (2)

- `LIST hosts`
- `FETCH host <hostname>`
- `LOOKUP hosts MATCHING <condition>`
- `... FILTER <condition>`
- `TIMESERIES <host>.<metric>`
  `START 2014-01-01 END 2014-12-31`

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

The SysDB Query Language – Example

```
sysdb=> LIST hosts FILTER :age < 2 * :interval;
[{
    "name": "monitor.lxc.tokkee.net",
    "last_update": "2014-04-03 10:26:41 +0200",
    "update_interval": "5m4s",
},{
    "name": "puppet.lxc.tokkee.net",
    "last_update": "2014-04-05 11:04:08 +0200",
    "update_interval": "5m2s"
}]
```


The SysDB Query Language – Example (2)

```
sysdb=> LOOKUP hosts WHERE attribute.architecture = 'amd64'
                           AND service.name =~ '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": [{...}]
}
```


The SysDB Query Language – Conditionals

Search and filter conditions:

```
MATCHING attribute.architecture = 'amd64'
         AND service.name =~ 'postgres'

MATCHING host =~ '\.tokkee\.net$'

FILTER :age >= 3 * :interval

FILTER :backend != 'mk-livestatus'
```


Use Cases

Use Cases

- More flexible web front-ends combining multiple back-ends
  - → correlate and annotate information
  - → central dashboard
  - → link to specialized front-ends
- Compare the back-ends (monitoring)
  - → Which hosts / services are missing in a back-end?
  - → Base for business-processes: What is the global status of all Windows systems in some data-center?
- Extend your CMDB with dynamic annotations or cross-checks

Future Directions

Future Directions

- Better and more integration with other systems
- Persistent store: RDBMS, graph database (?)
- Interface to query other live data (e.g., monitoring status) from back-ends
- Distributed architecture (HA and load-balancing)
- Extensible Web-Interface
- Extend the type system and filters
- ...

SysDB – System DataBase

Thank you for your attention!

Questions, comments, rants?

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

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

@SystemDatabase