System integration and analysis

By José M. Román

FiberCli

We are pure fiber

JOSE MANUEL ROMAN

17 years experience, Mikrotik Certified Consultant and Trainer. MTCNA, MTCRE, MTCTCE, MTCUME, MTCWE, MTCINE, CISA, CISSP, Master ITIL

- (Now) CEO at Fibercli
- (2015 – Now) CEO @ WISP Cloud Networking Spain
- (2008 – Now) Security Consultant and Analyst
- ***(2000 – 2007) Networking, security and itil teacher .***

MADRID / PRAGUE

☎ +34 652 241431

@MAFIASOLEH

Jose.roman@fibercli.com

FAJAR NUGROHO

Network Engineer by Job and Troublemaker by Act, currently focusing on MikroTik, Juniper, Arista, UBNT, Vmware Virtualization, Linux/Unix (Debian & FreeBSD). CCNA, MTCNA, MTCRE, MTCTCE, JNCIA, JNCIS-ENT, JNCIS-SP, JNCIP-SP, MikroTik Certified Trainer.

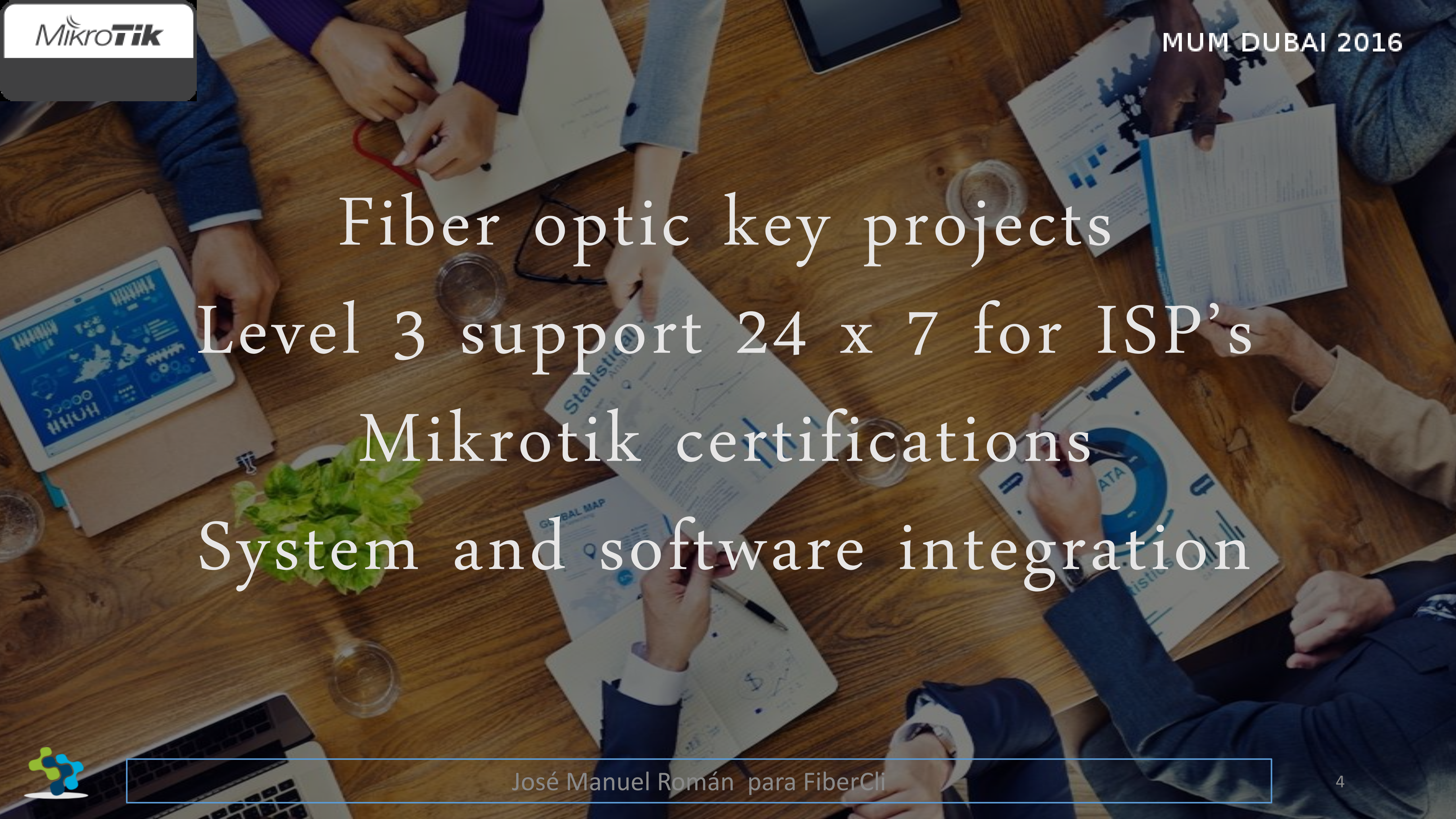
- (2015 – 2016) Infrastucture *(System, Network & Security)* Engineer. @ Technology and Information Department of Jakarta Capital City and Jakarta SmartCity
- (2012 – Now) Freelancer @ SMB to Enterprise customers
- (2008 – 2012) Helpdesk, NOC *(Network Operator Center).* @ Wireless Internet Service Provider and Triple Play (CaTV, VoIP and Internet) Service Provider

TOLEDO / JAKARTA

☎ +62 813 1777 1455

@MAFIASOLEH

fajar@fibercli.com



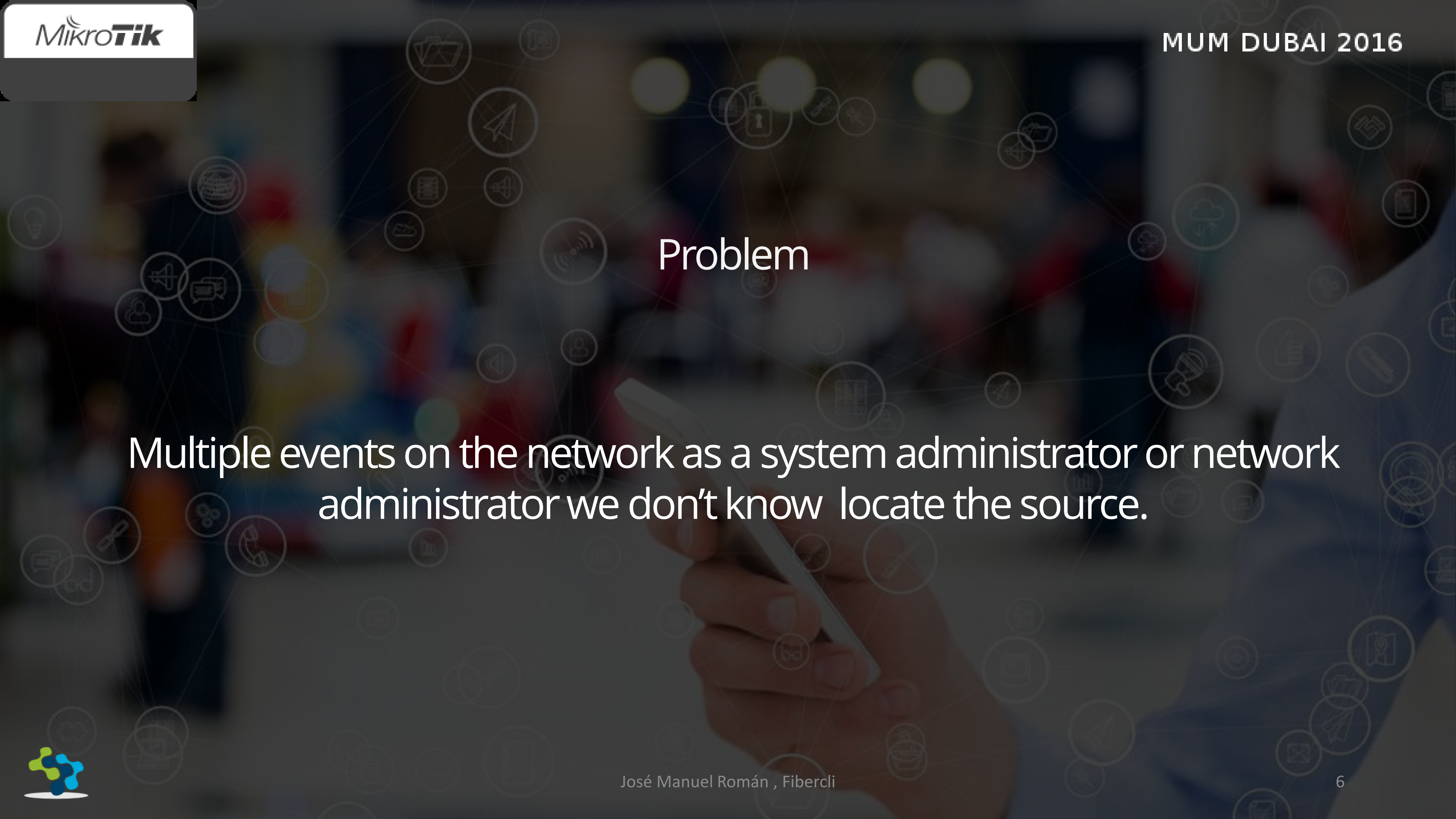
Fiber optic key projects

Level 3 support 24 x 7 for ISP’s

Mikrotik certifications

System and software integration

20% Disccount for MUM assistant



Problem

Multiple events on the network as a system administrator or network administrator we don’t know locate the source.

Symptom

Multiple incidents that are not managed.

Feeling of lack of control over the network.

Solution

Centralized system to collect, normalize, visualization and analysis

AGENDA

- Intro
- Architecture
- ELK (ElasticSearch, Logstash, Kibana)
- Mkt + AAA+ with Freeradius and DB centralized
- Mkt + Centralized Log and ELK
- Mkt + Monitoring and ELK
- Mkt + Netflow and ELK
- Q and A

¿What is?

ELK

Elasticsearch

Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

Elasticsearch

Elasticsearch is developed in Java and is released as open source under the terms of the Apache License

Elasticsearch

- Distributed, scalable, and highly available
- Real-time search and analytics capabilities
- Sophisticated RESTful API

- https://www.elastic.co/products/elasticsearch

Elasticsearch

- **Schema-free, REST & JSON based distributed search engine**
- **Open Source: Apache License 2.0**
- **Easy to understand, yet very powerful query language**
  - *Full text search (phrase, fuzzy)*
  - *Numeric search (support ranges, dates, ipv4 addresses)*
  - *Highlighting*
  - *Aggregations*
  - *Suggestions*

Logstash

Logstash is a tool to collect, process, and forward events and log messages. Collection is accomplished via configurable input plugins including raw socket/packet communication, file tailing, and several message bus clients.

Logstash

- Centralize data processing of all types
- Normalize varying schema and formats
- Quickly extend to custom log formats
- Easily add plugins for custom data sources

- https://wikitech.wikimedia.org/wiki/Logstash

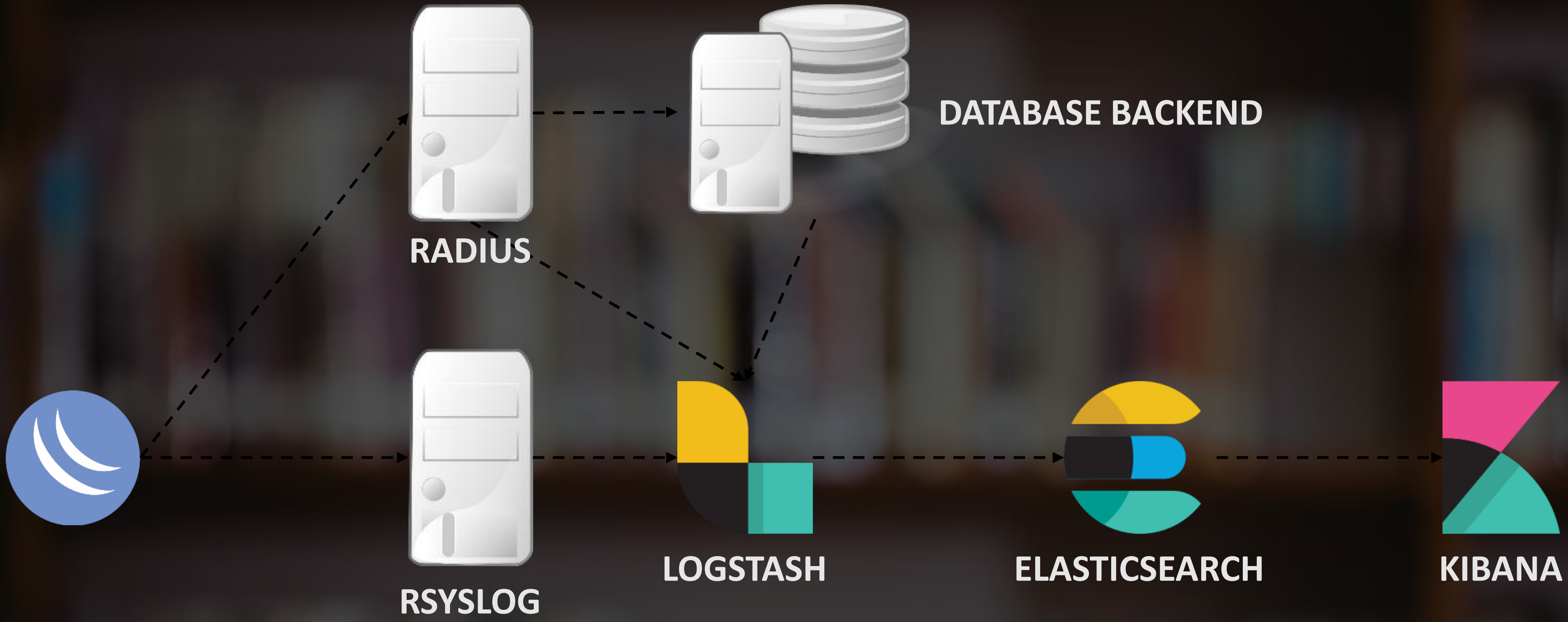
Logstash

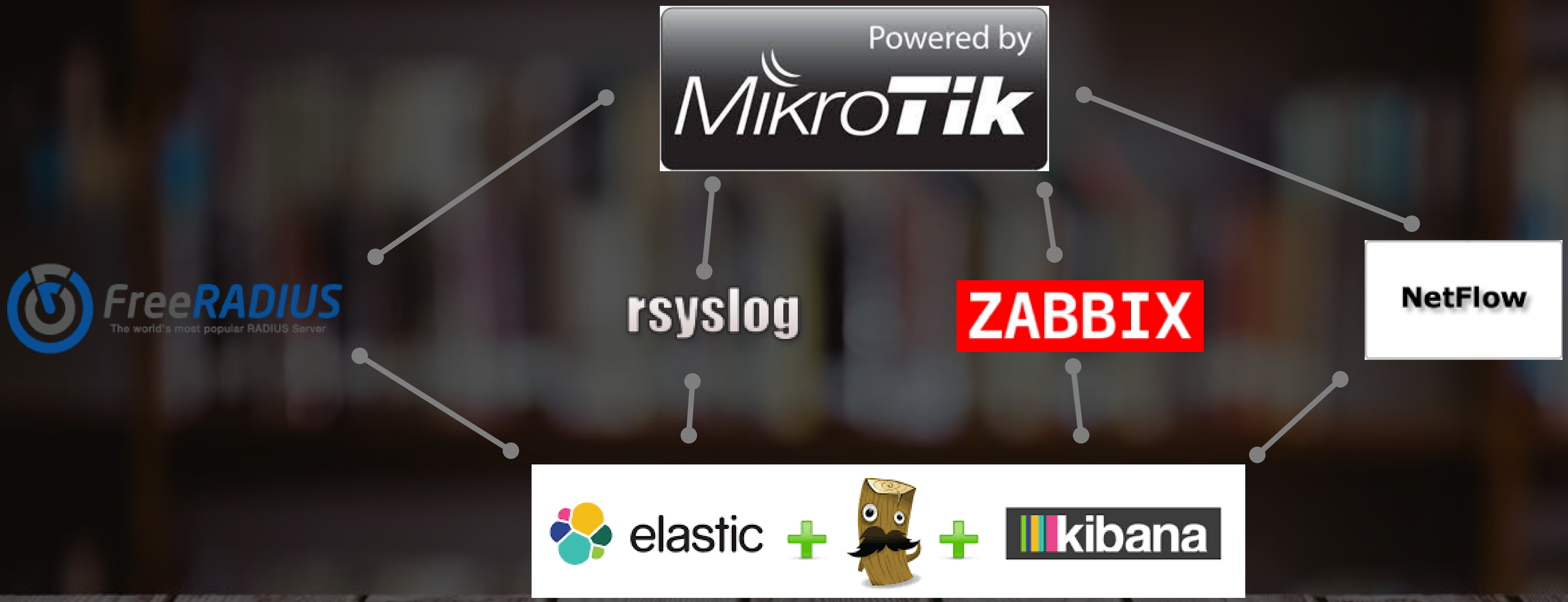
- Inputs: collect data from variety of sources
- Filters: parse, process and enrich data
- Outputs: push data to a variety of destinations

Kibana

- Execute queries on your data & visualize results
- Add/remove widgets
- Share/Save/Load dashboards
- Open Source: Apache License 2.0

Big Picture


RADIUS
DATABASE BACKEND
RSYSLOG
LOGSTASH
ELASTICSEARCH
KIBANA


Powered by
MikroTik
FreeRADIUS
The world's most popular RADIUS Server
rsyslog
ZABBIX
NetFlow
elastic
+
+
kibana

AAA System

Radius

RADIUS is an application level protocol that carries Authentication, Authorization and Accounting (AAA) configuration information between a Network Access Server (NAS) and a Shared Authentication Server. Radius defined in RFC 2865

Radius

In MikroTik RouterOS itself support RADIUS for Hotspot, PPP, DHCP, Wireless and Login. RADIUS using transport protocol UDP.

UDP Port 1812 – Authentication

UDP Port 1813 – Accounting

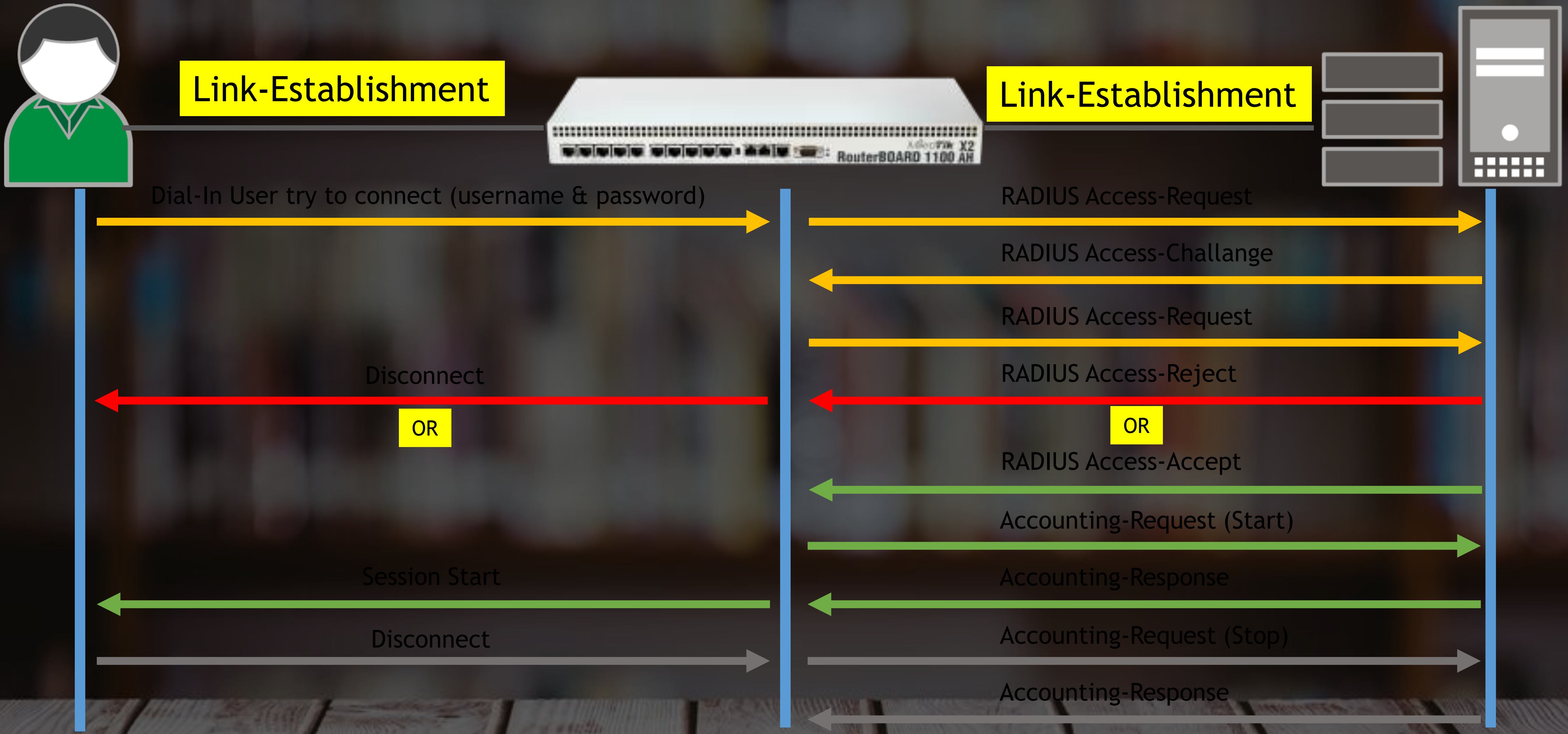
Radius

RADIUS operation typically split into three type :

- Dial-In User : User who requesting for login and password
- Network Access Server (NAS) / RADIUS Client : Device who accept the request from dial-in user and

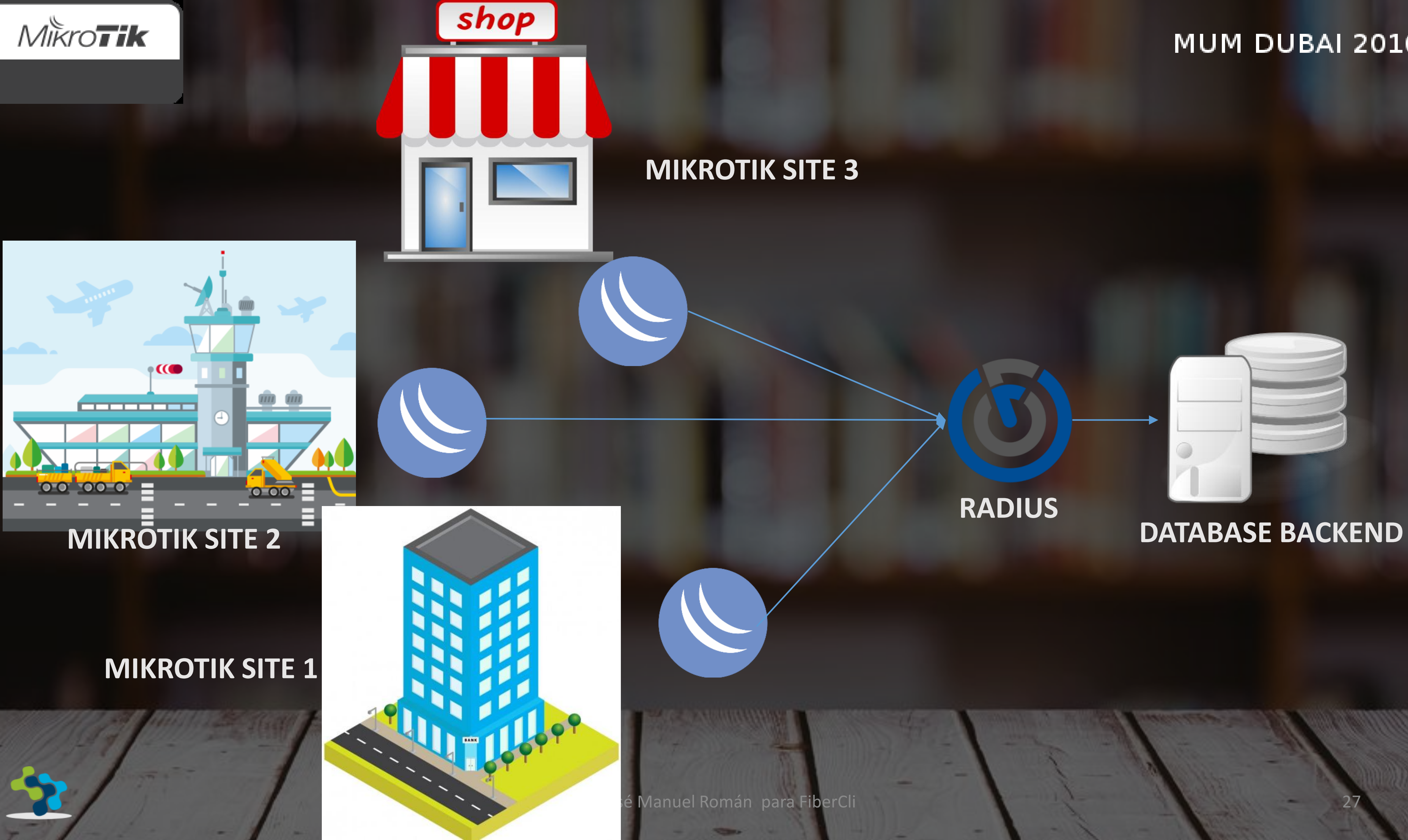
  : Forward into RADIUS Server.
- Shared Authentication Server / RADIUS Server : Device who make a decision for request (Accept,

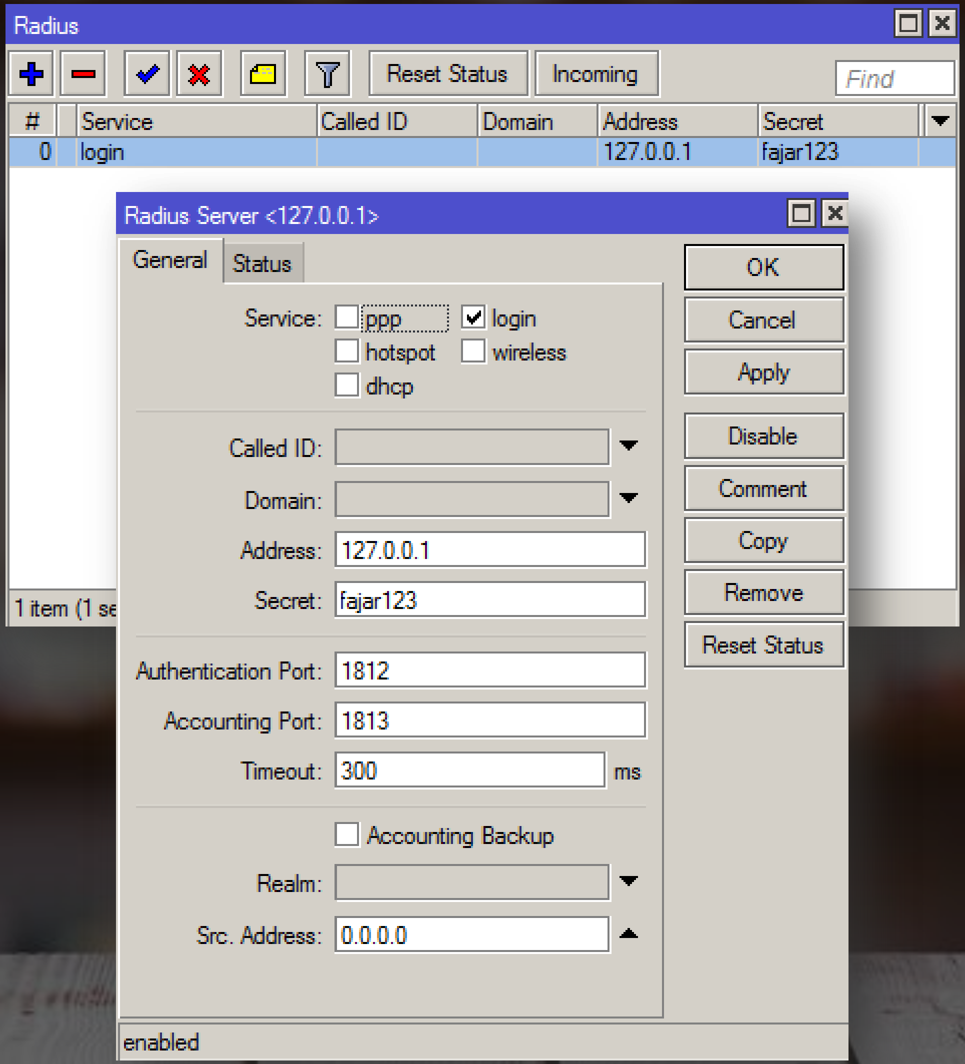
  : Reject or Challenge)



Link-Establishment
Link-Establishment
RouterBOARD 1100 AH
Dial-In User try to connect (username & password)
RADIUS Access-Request
RADIUS Access-Challange
RADIUS Access-Request
Disconnect
RADIUS Access-Reject
OR
OR
RADIUS Access-Accept
Accounting-Request (Start)
Session Start
Accounting-Response
Disconnect
Accounting-Request (Stop)
Accounting-Response

Topology


shop
MIKROTIK SITE 3
MIKROTIK SITE 2
MIKROTIK SITE 1
BANK
RADIUS
DATABASE BACKEND



#	Service	Called ID	Domain	Address	Secret
0	login			127.0.0.1	fajar123

1 item (1 se

Radius Server <127.0.0.1>

Service: □ ppp ☑ login □ hotspot □ wireless □ dhcp

Called ID:

Domain:

Address: 127.0.0.1

Secret: fajar123

Authentication Port: 1812

Accounting Port: 1813

Timeout: 300 ms

□ Accounting Backup

Realm:

Src. Address: 0.0.0.0

enabled

service *(ppp|login|hotspot|wireless|dhcp; Default:)*

- *hotspot* - HotSpot authentication service
- *login* - router's local user authentication
- *ppp* - Point-to-Point clients authentication
- *wireless* - wireless client
- *dhcp* - DHCP protocol client authentication

Address *(IPv4/IPv6 address; Default: 0.0.0.0)* IPv4 or IPv6 address of RADIUS server.

Secret *(string; Default:)* Shared secret used to access the RADIUS server.

Freeradius produces several logs that we can process with logstash

http://code.metager.de/source/xref/freeradius/server/doc/schemas/logstash/

```
#       Tue Mar 10 15:32:24 2015
12#             Packet-Type = Access-Request
13#             User-Name = "test@example.com"
14#             Calling-Station-Id = "01-02-03-04-05-06"
15#             Called-Station-Id = "aa-bb-cc-dd-ee-ff:myssid"
16#             NAS-Port = 10
17#             NAS-IP-Address = 10.9.0.4
18#             NAS-Identifier = "Wireless-Controller-1"
19#             Service-Type = Framed-User
20#             NAS-Port-Type = Wireless-802.11
21#
```

http://code.metager.de/source/xref/freeradius/server/doc/schemas/logstash/

```
input {
  stdin {
    type => radiusdetail
  }
}
```

http://code.metager.de/source/xref/freeradius/server/doc/schemas/logstash/

```
filter {

  if [type] == "radiusdetail" {

          # join all lines of a record together
          multiline {
                  pattern => "^[^\t]"
                  negate => true
                  what => "previous"
          }

          # pull off the timestamp
          grok {
                  match => [ "message", "^(?<timestamp>[^\n\t]+)[\n\t]" ]
          }

          # create the timestamp field
          date {
                  match => [ "timestamp", "EEE MMM dd HH:mm:ss yyyy",
                                                      "EEE MMM  d HH:mm:ss yyyy" ]

          }

          # split the attributes and values into fields
          kv {
                  field_split => "\n"
                  source => "message"
                  trim => "\" "
                  trimkey => "\t "
          }
  }
}
```

http://code.metager.de/source/xref/freeradius/server/doc/schemas/logstash/

```
output {
 if [type] == "radiusdetail" {
  elasticsearch {
    host => localhost
    protocol => http
    cluster => elasticsearch
    index_type => "detail"
    index => "radius-%{+YYYY.MM.dd}"
    flush_size => 1000
  }
 }
}
```

http://code.metager.de/source/xref/freeradius/server/doc/schemas/logstash/

Centralize Log

RSYSLOG

RSYSLOG stand for "the rocket-fast system for log processing“ is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. It implements the basic syslog protocol, extends it with content-based filtering, rich filtering capabilities, flexible configuration options and adds features such as using TCP for transport

http://www.rsyslog.com/rsyslog-8-19-0-v8-stable-released/

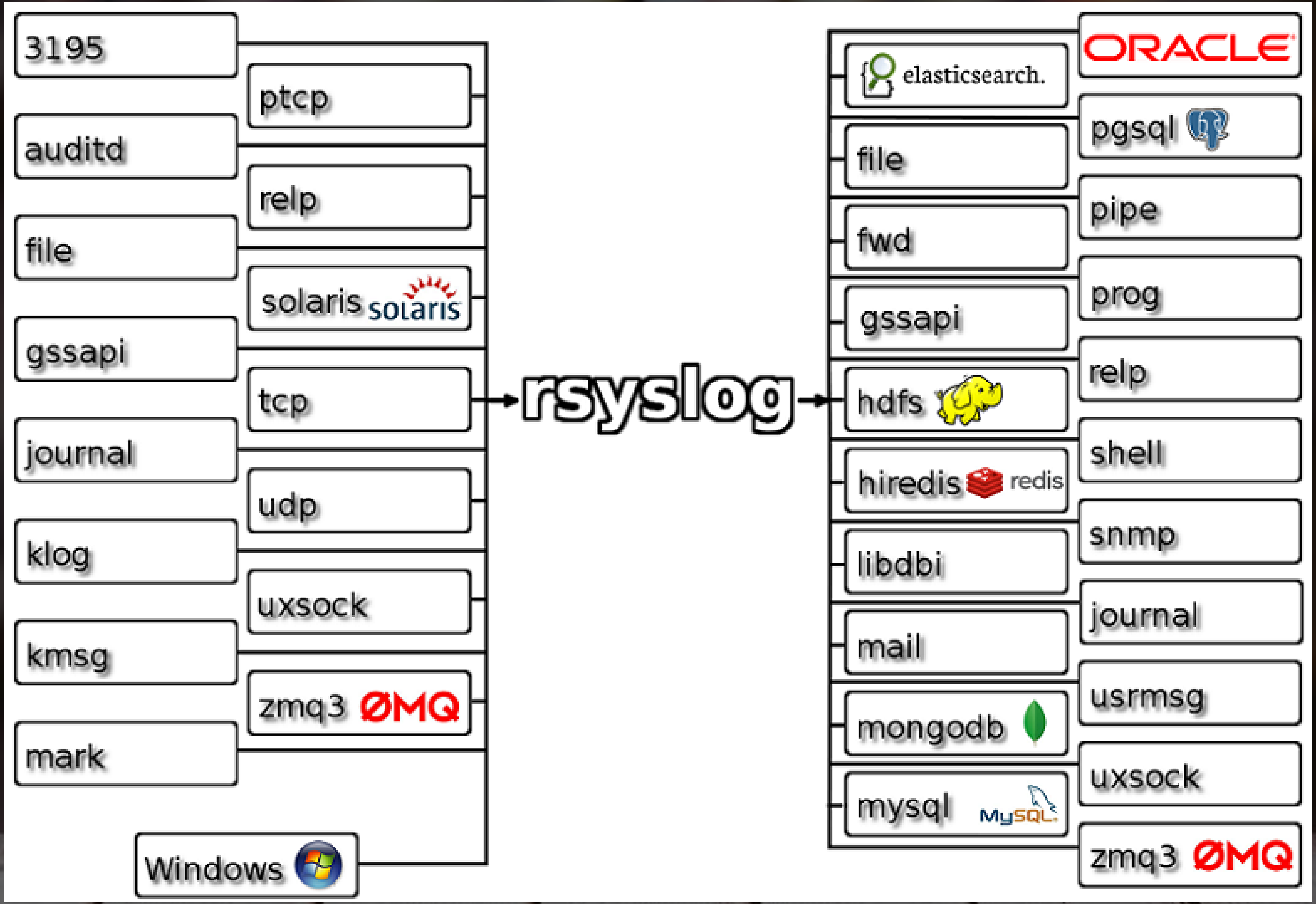
RSYSLOG

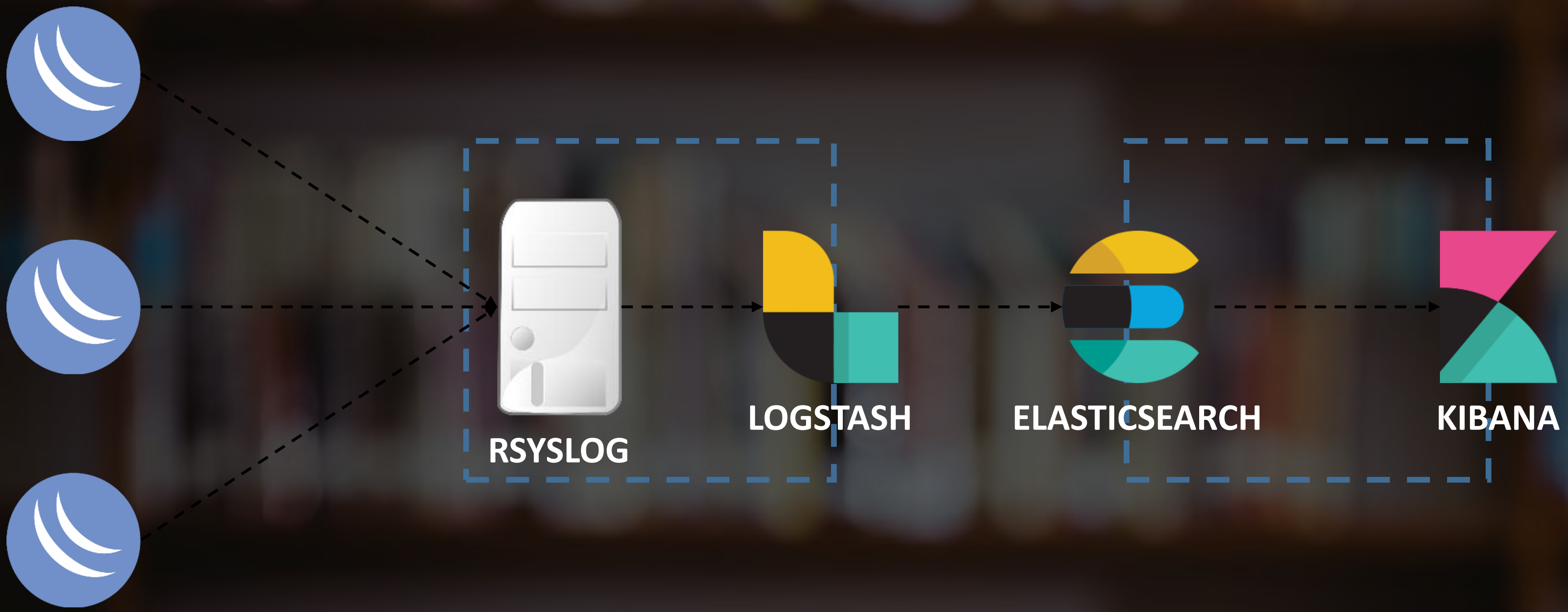
- Protocol supported by rsyslog are:
- ISO 8601 timestamp with millisecond granularity and timezone information
- The addition of the name of relays in the host fields to make it possible to track the path a given message has traversed
- Reliable transport using TCP
- Support GSS-API and TLS

RSYSLOG

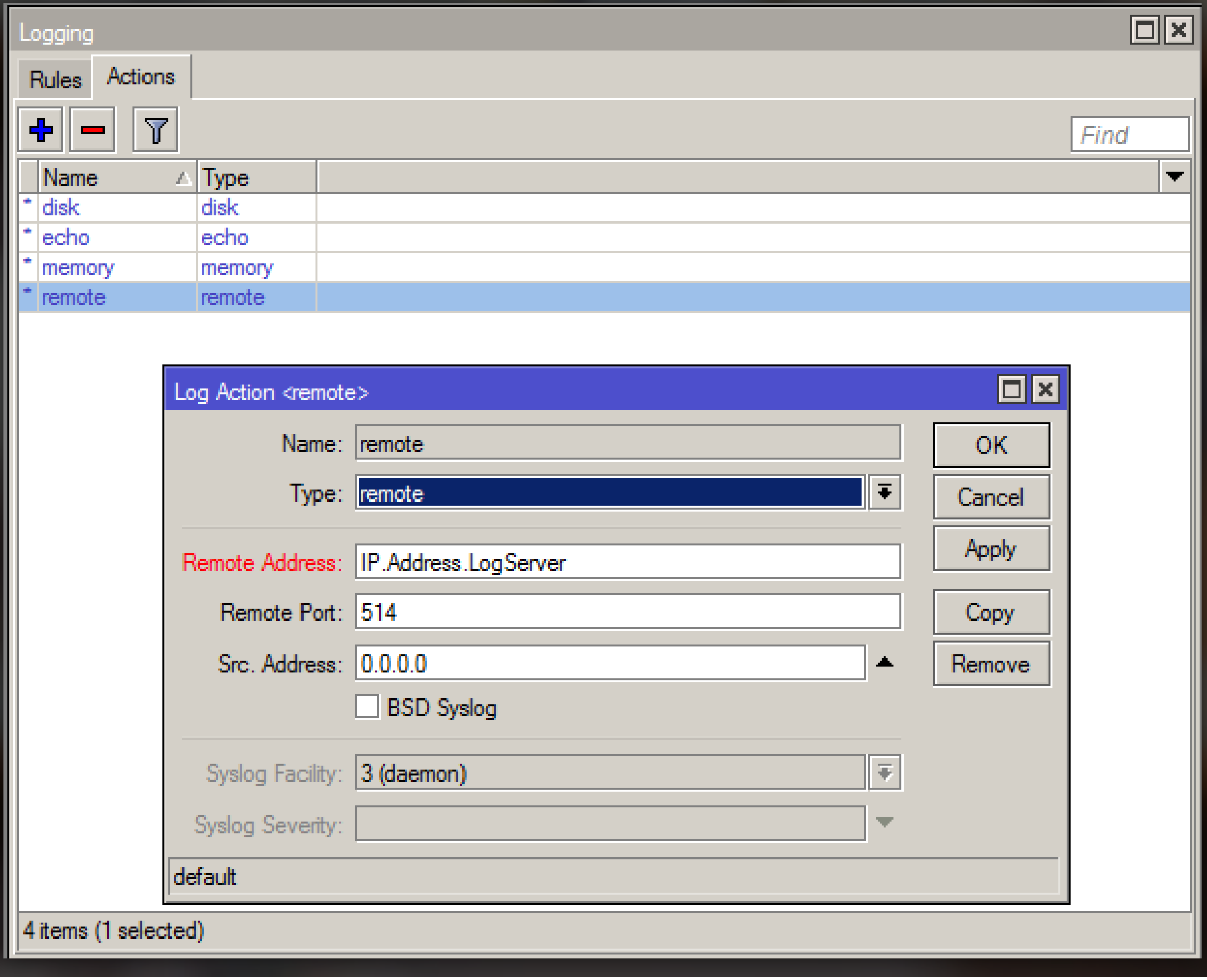
- Logging directly into various database engines.
- Support for RFC 5424, RFC 5425, RFC 5426
- Complete input/output support for systemd journal

Topology


3195
ptcp
auditd
relp
file
solaris solaris
gssapi
tcp
journal
udp
klog
uxsock
kmsg
zmq3 ØMQ
mark
Windows
rsyslog
elasticsearch.
ORACLE
pgsql
file
pipe
fwd
prog
gssapi
relp
hdfs
shell
hiredis redis
snmp
libdbi
journal
mail
usrmsg
mongodb
uxsock
mysql MySQL
zmq3 ØMQ


RSYSLOG
LOGSTASH
ELASTICSEARCH
KIBANA

MikroTik log Configuration



Name	Type
* disk	disk
* echo	echo
* memory	memory
* remote	remote

Log Action <remote>

Name: remote

Type: remote

Remote Address: IP.Address.LogServer

Remote Port: 514

Src. Address: 0.0.0.0

□ BSD Syslog

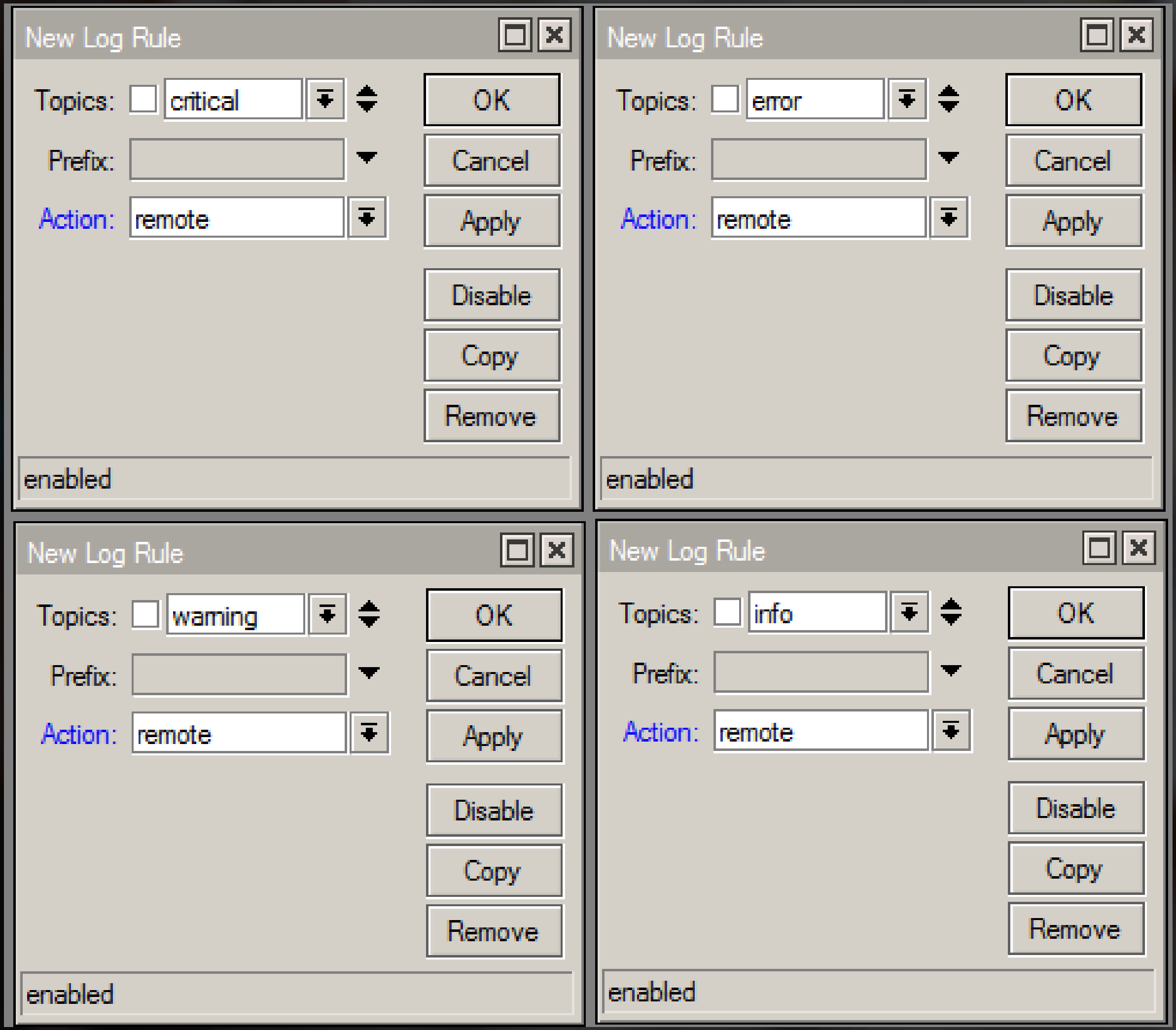
Syslog Facility: 3 (daemon)

Syslog Severity:

default

4 items (1 selected)

```
system logging action set remote remote=ip.address.log.server
```



New Log Rule

Topics: □ critical

Prefix:

Action: remote

enabled

New Log Rule

Topics: □ error

Prefix:

Action: remote

enabled

New Log Rule

Topics: □ warning

Prefix:

Action: remote

enabled

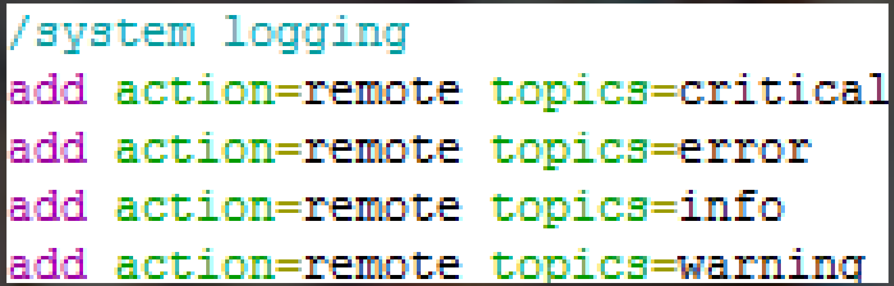
New Log Rule

Topics: □ info

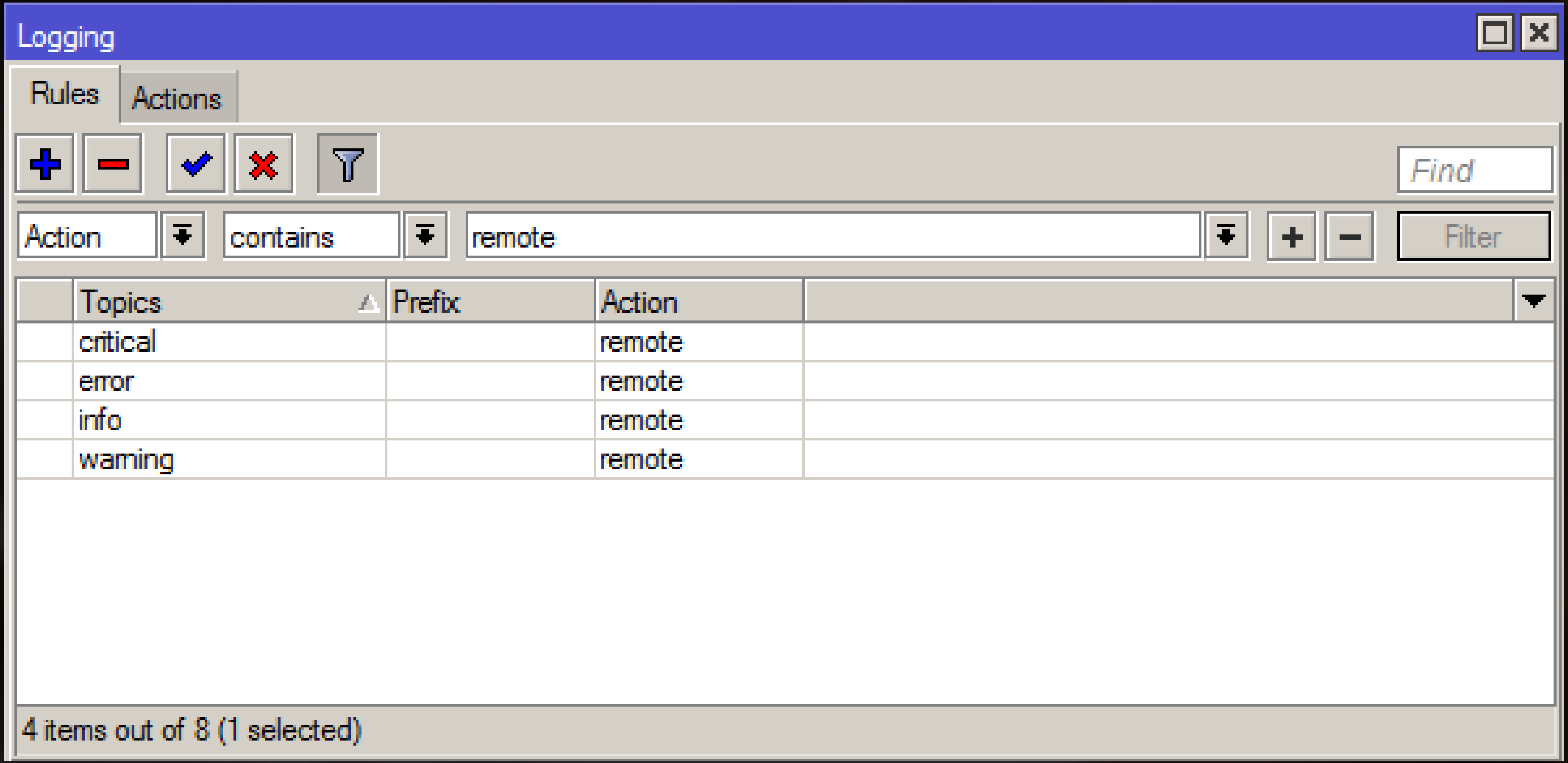
Prefix:

Action: remote

enabled



```
/system logging
add action=remote topics=critical
add action=remote topics=error
add action=remote topics=info
add action=remote topics=warning
```



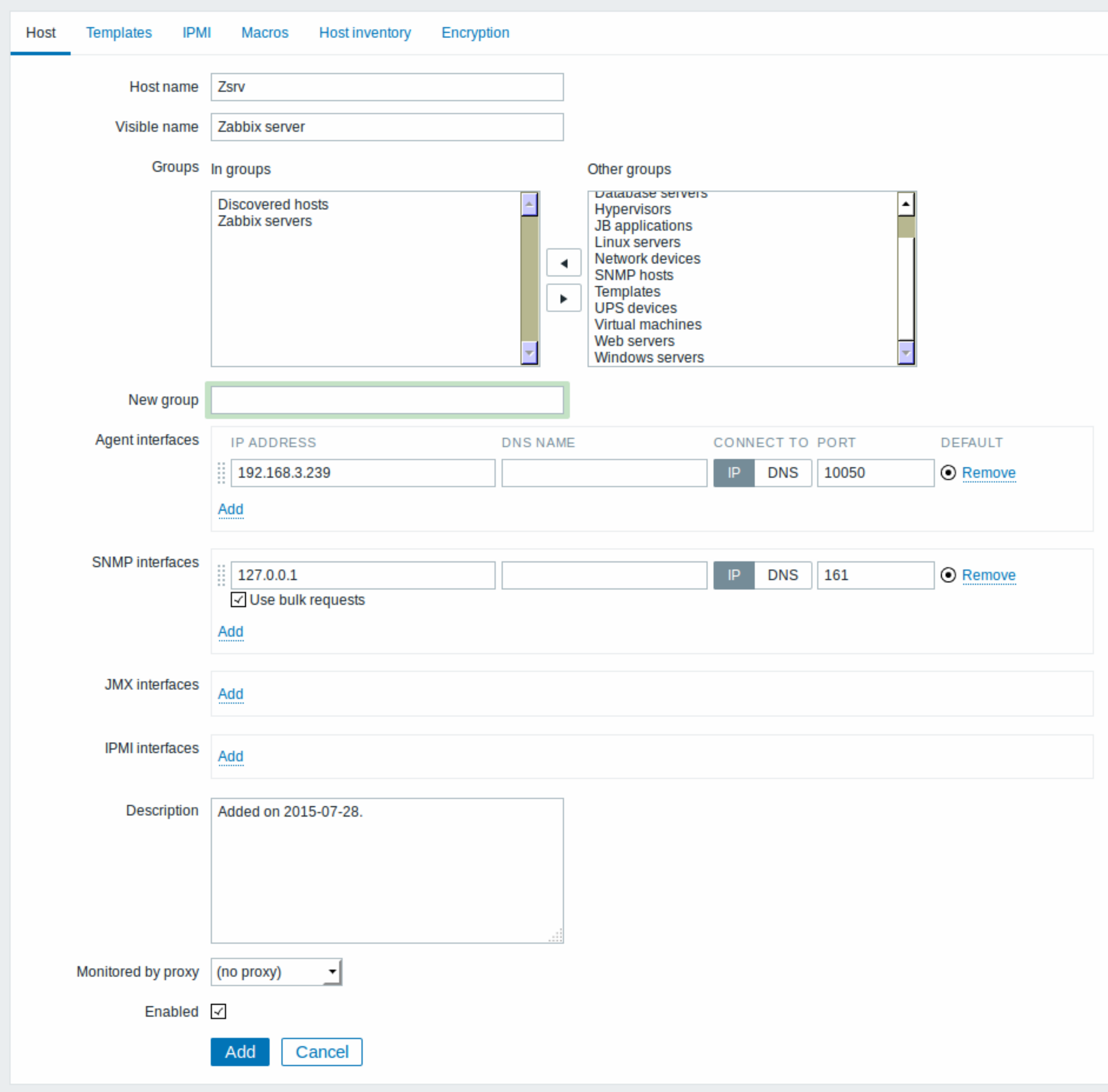
Action contains remote

Topics	Prefix	Action
critical		remote
error		remote
info		remote
warning		remote

4 items out of 8 (1 selected)

Monitoring

```
/snmp set enabled=yes contact="jose.roman@fibercli.com" location="Mum Madrid" trap-community=public trap-version=2
```



Host name: Zsrv

Visible name: Zabbix server

Groups

In groups	Other groups
Discovered hosts Zabbix servers	Database servers Hypervisors JB applications Linux servers Network devices SNMP hosts Templates UPS devices Virtual machines Web servers Windows servers

New group:

Agent interfaces

IP ADDRESS	DNS NAME	CONNECT TO	PORT	DEFAULT
192.168.3.239		IP DNS	10050	● Remove

SNMP interfaces

IP ADDRESS	DNS NAME	CONNECT TO	PORT	DEFAULT
127.0.0.1		IP DNS	161	● Remove

☑ Use bulk requests

JMX interfaces: Add

IPMI interfaces: Add

Description: Added on 2015-07-28.

Monitored by proxy: (no proxy)

Enabled: ☑

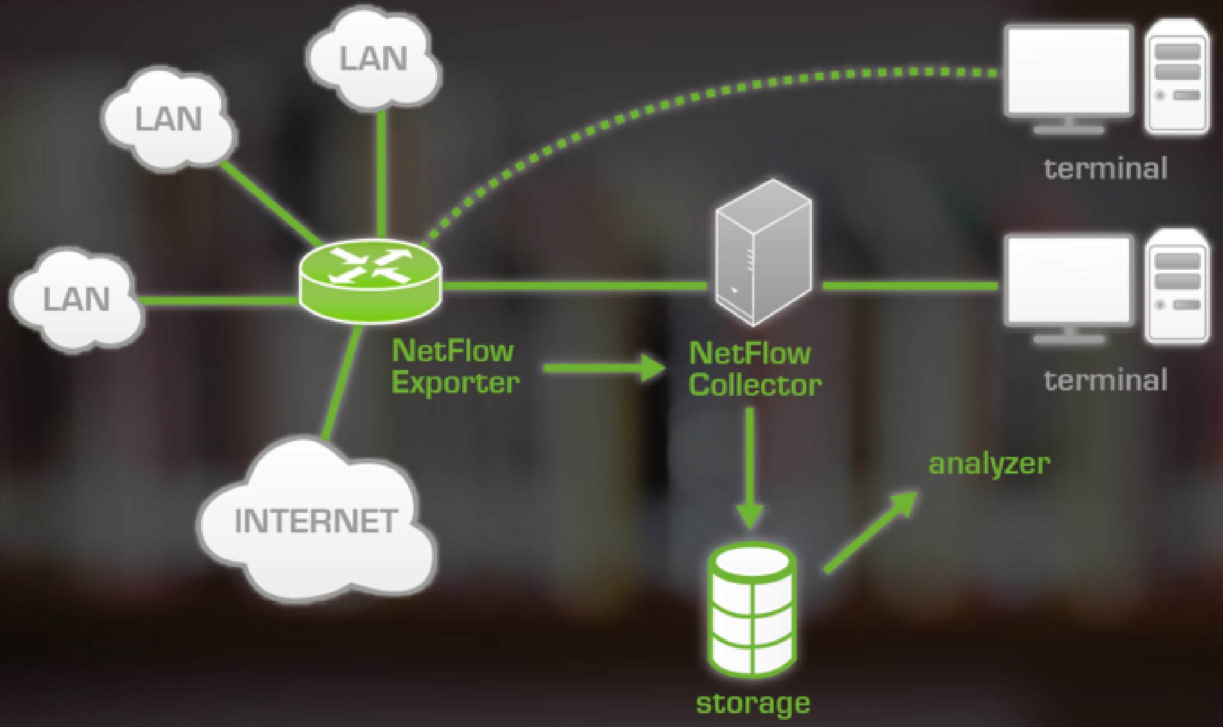
There are clients to export data to databases like fluentdb.

https://github.com/jojohappy/zabbix-relay

We can integrate Zabbix events as input in Logstach, with the goal to have a decoupled monitorization.

Netflow

NetFlow is a network protocol created by Cisco Systems to collect information about ip traffic.


LAN
LAN
LAN
INTERNET
NetFlow Exporter
NetFlow Collector
terminal
terminal
storage
analyzer

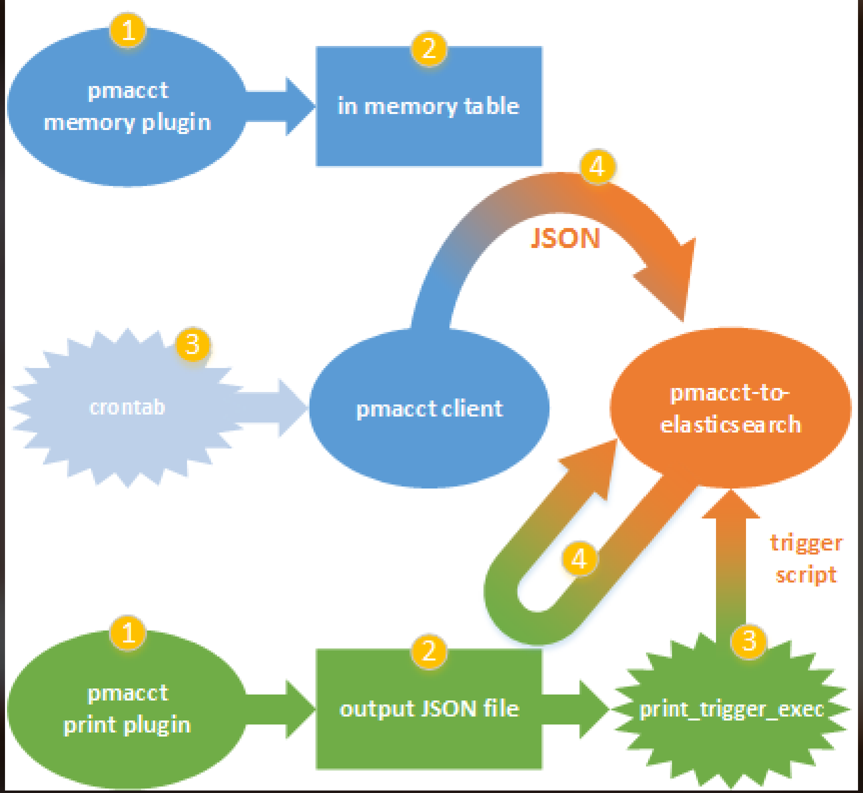
```
/ip traffic-flow  set active-flow-timeout=30m cache-entries=1M
\enabled=yes  inactive-flow-timeout=15s interfaces=all
```

```
/ip traffic-flow target add dst-address=ip.server port=5055 disabled=no
\v9-template-refresh=20 v9-template-timeout=30m  version=9
```

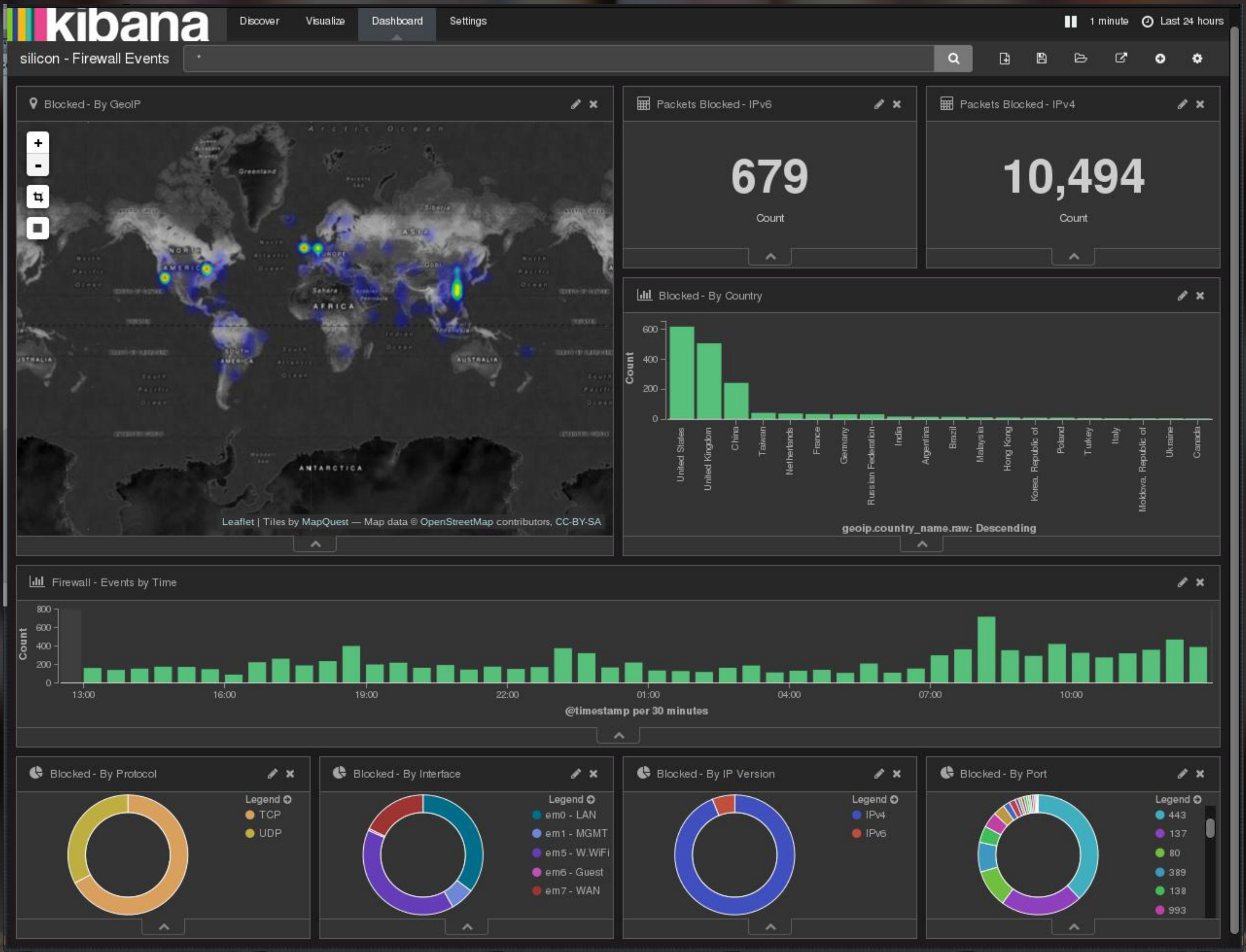
To collect the output we need a netflow collector, for example pmacctsonda.

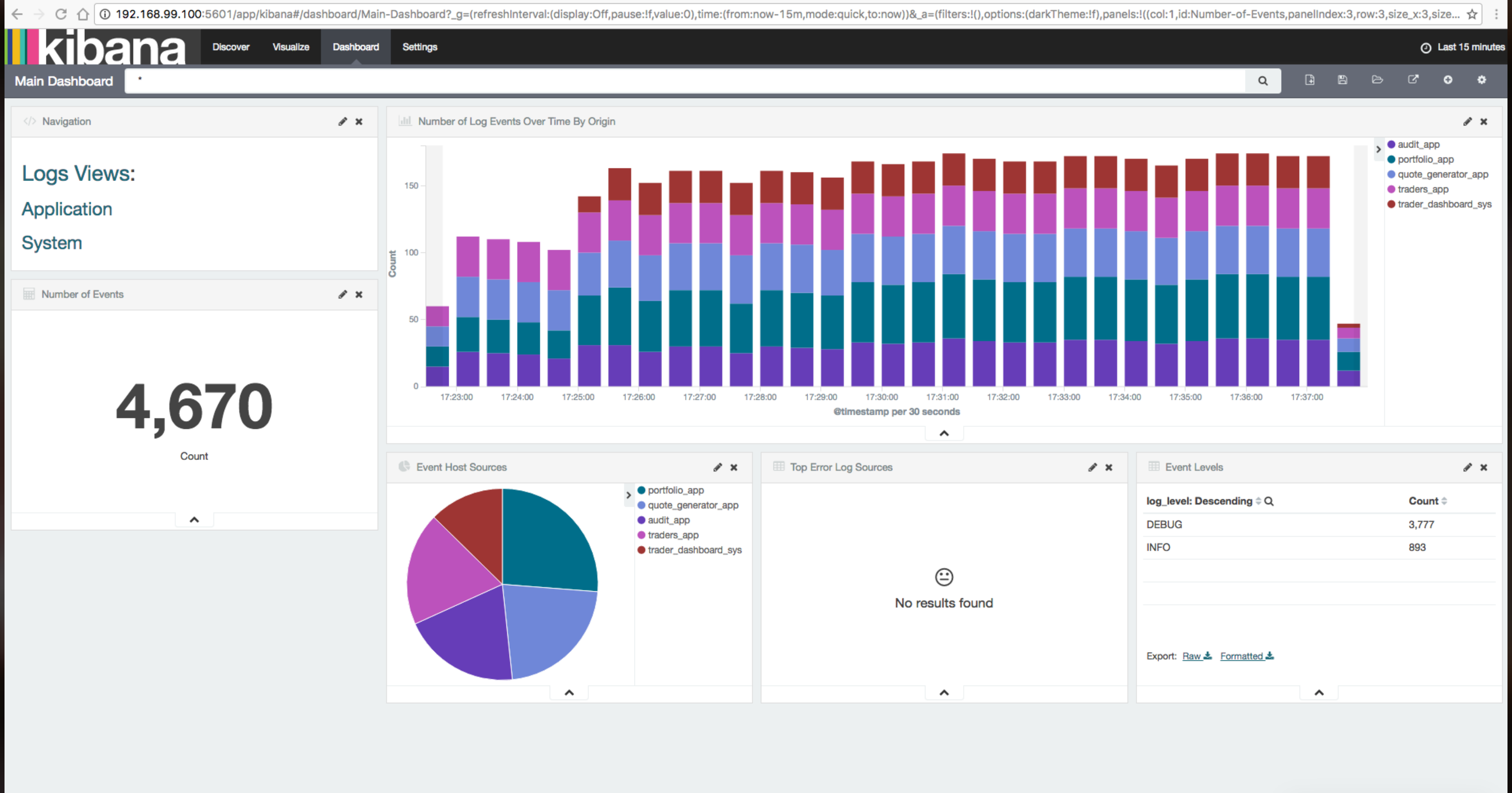
When we collect the logs with pcmacct we'll send the output in json format to ElasticSearch.

https://github.com/pierky/pmacct-to-elasticsearch/blob/master/CONFIGURATION.md



https://github.com/pierky/pmacct-to-elasticsearch/blob/master/CONFIGURATION.md


kibana
Discover
Visualize
Dashboard
Settings
1 minute
Last 24 hours
silicon - Firewall Events
*
Blocked - By GeoIP
Leaflet | Tiles by MapQuest — Map data © OpenStreetMap contributors, CC-BY-SA
Packets Blocked - IPv6
679
Count
Packets Blocked - IPv4
10,494
Count
Blocked - By Country
Count
600
400
200
0
United States
United Kingdom
China
Taiwan
Netherlands
France
Germany
Russian Federation
India
Argentina
Brazil
Malaysia
Hong Kong
Korea, Republic of
Poland
Turkey
Italy
Moldova, Republic of
Ukraine
Canada
geoip.country_name.raw: Descending
Firewall - Events by Time
Count
800
600
400
200
0
13:00
16:00
19:00
22:00
01:00
04:00
07:00
10:00
@timestamp per 30 minutes
Blocked - By Protocol
Legend
TCP
UDP
Blocked - By Interface
Legend
em0 - LAN
em1 - MGMT
em5 - W.WiFi
em6 - Guest
em7 - WAN
Blocked - By IP Version
Legend
IPv4
IPv6
Blocked - By Port
Legend
443
137
80
389
138
993



192.168.99.100:5601/app/kibana#/dashboard/Main-Dashboard?_g=(refreshInterval:(display:Off,pause:!f,value:0),time:(from:now-15m,mode:quick,to:now))&_a=(filters:!(),options:(darkTheme:!f),panels:!((col:1,id:Number-of-Events,panelIndex:3,row:3,size_x:3,size...

Main Dashboard

Navigation

Logs Views:

Application

System

Number of Events

4,670

Count

Number of Log Events Over Time By Origin

Event Host Sources

Top Error Log Sources

No results found

Event Levels

log_level: Descending	Count
DEBUG	3,777
INFO	893

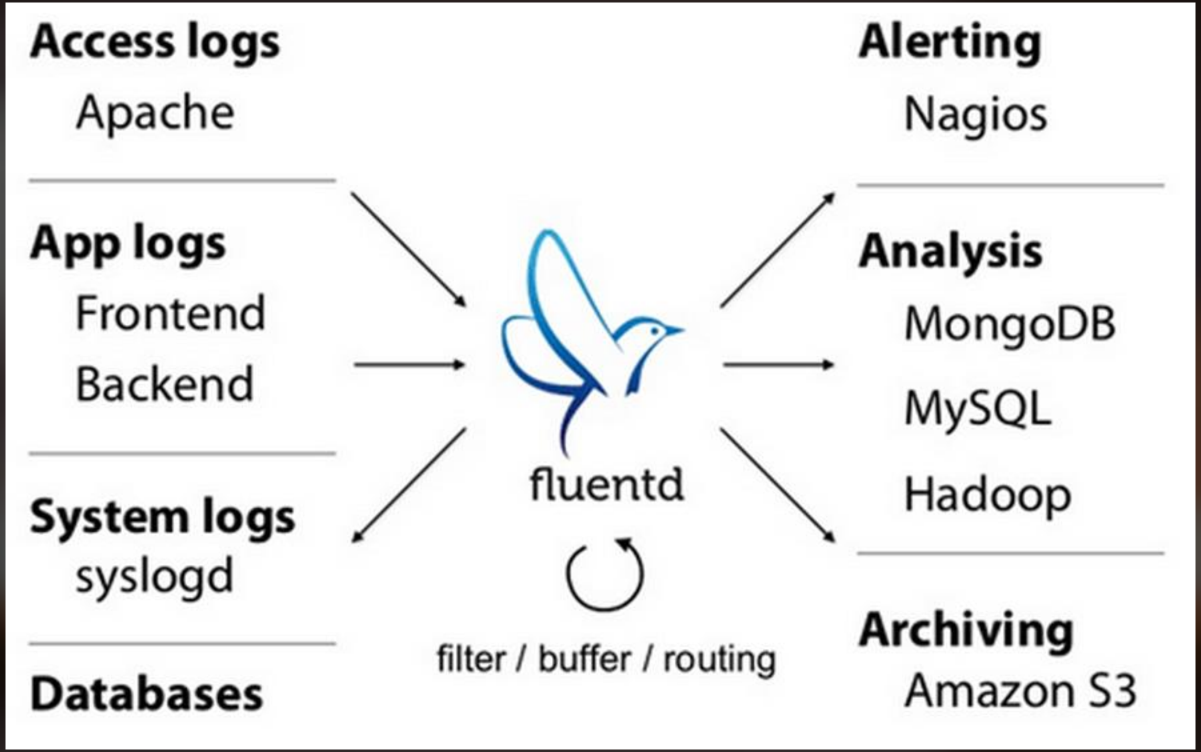
Export: Raw Formatted

Additional resources

Grafana

Influxdb

Fluentd



Access logs
Apache

App logs
Frontend
Backend

System logs
syslogd

Databases

fluentd

filter / buffer / routing

Alerting
Nagios

Analysis
MongoDB
MySQL
Hadoop

Archiving
Amazon S3

Thank you

jose.roman@fibercli.com

www.fibercli.com