Kea

A modern DHCP engine

UKNOF40

Tomek Mrugalski
tomek(at)isc(dot)org

© 2018 ISC

If you never heard about Kea…

- Modern DHCPv4, DHCPv6 and DDNS servers
- Performance
- Scalable (millions of devices)
- On-line reconfiguration (no restarts after config changes)
- Feature rich: shared networks, v6, PD, custom options,…
- **Database Backends**
- **Hooks** (3rd party libraries, like apache modules)
- **REST** management API
- Linux, BSDs, MacOS, …
- Open source (MPL2)
- 1.4.0 beta about to be released (May 2018)

© 2018 ISC

Backends

- Leases, host reservations in DB (1.4)
  - CSV
  - MySQL, PostgreSQL
  - Cassandra
- Configuration in DB likely in 1.5
- SQL data can be modified any time
- All changes applied instantly (no restart)
- Can manipulate the DB directly or
- Use host commands (1.2) and subnets (1.3)

© 2018 ISC

Hooks (1 of 2)

- 1.1: **User Check** – example access control
- 1.2: **Forensic Logging** – audit trail for legal purposes
- 1.2: **Flexible Identifier** – identify hosts by expression, e.g. concat(relay4[2].hex, relay4[6].hex)
- 1.2: **Host Commands** – query, add and delete host reservations using REST interface
- 1.3: **Subnet management** (add, get, update, delete subnets and shared networks via REST API)
- 1.3: **Extra lease commands** (add, get, update, delete, wipe all, get all leases via REST API)

Open source

Premium

Hooks (2 of 2)

- 1.4: **HA** – high availability solution (heartbeat, failure detection, lease updates, recovering DB from partner)
- 1.4: **Radius** – access control and host reservation using FreeRadius, accounting
- 1.4: **Host Caching** – cache host responses locally from slower backends for extra performance (includes negative caching)
- 1.5: **Limits** - ability to rate limit queries, limit # of leases per subnet, pool, port, device, time restricted leases (valid until, not valid before)

Open source

Premium

Anyone can write hooks

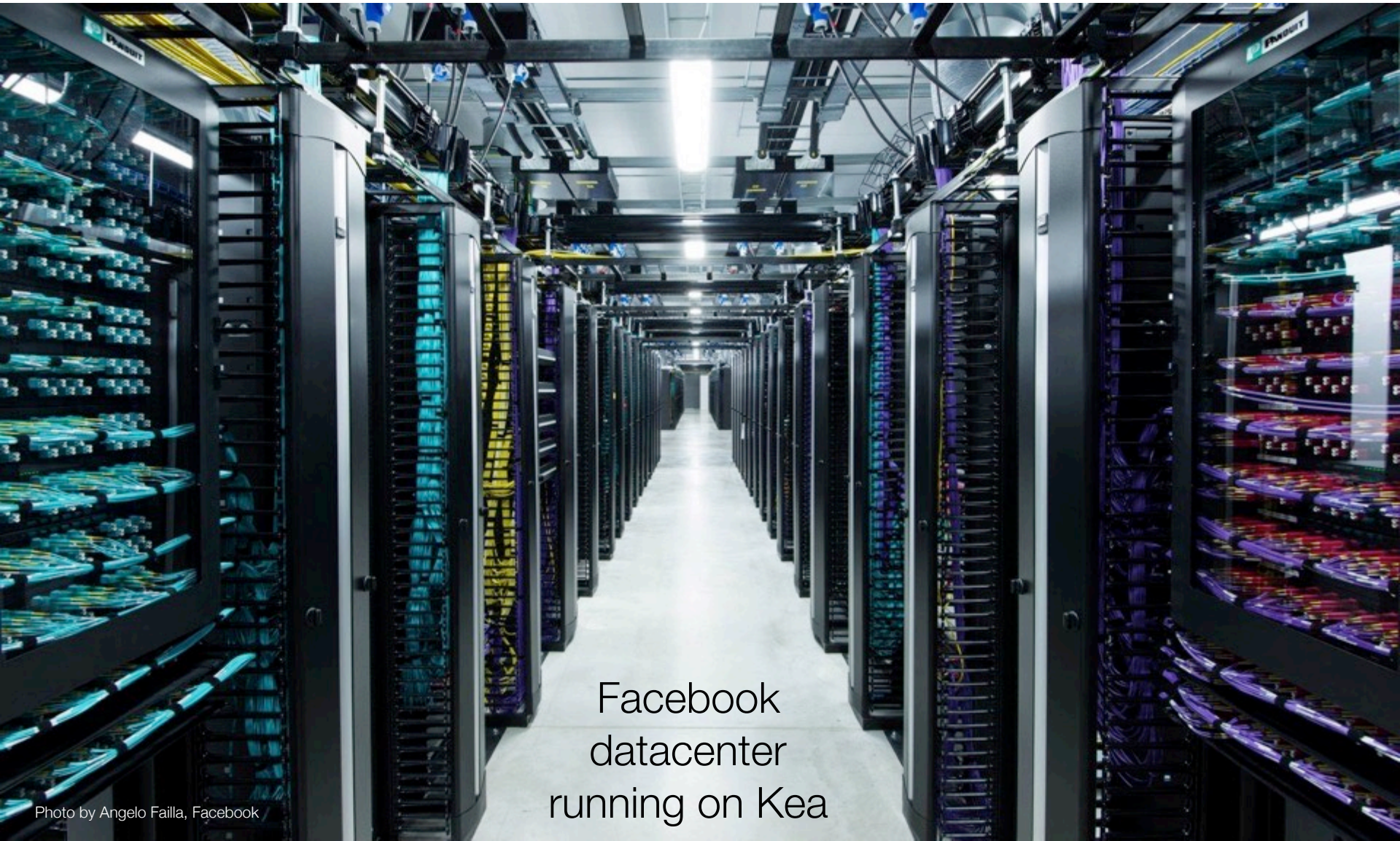


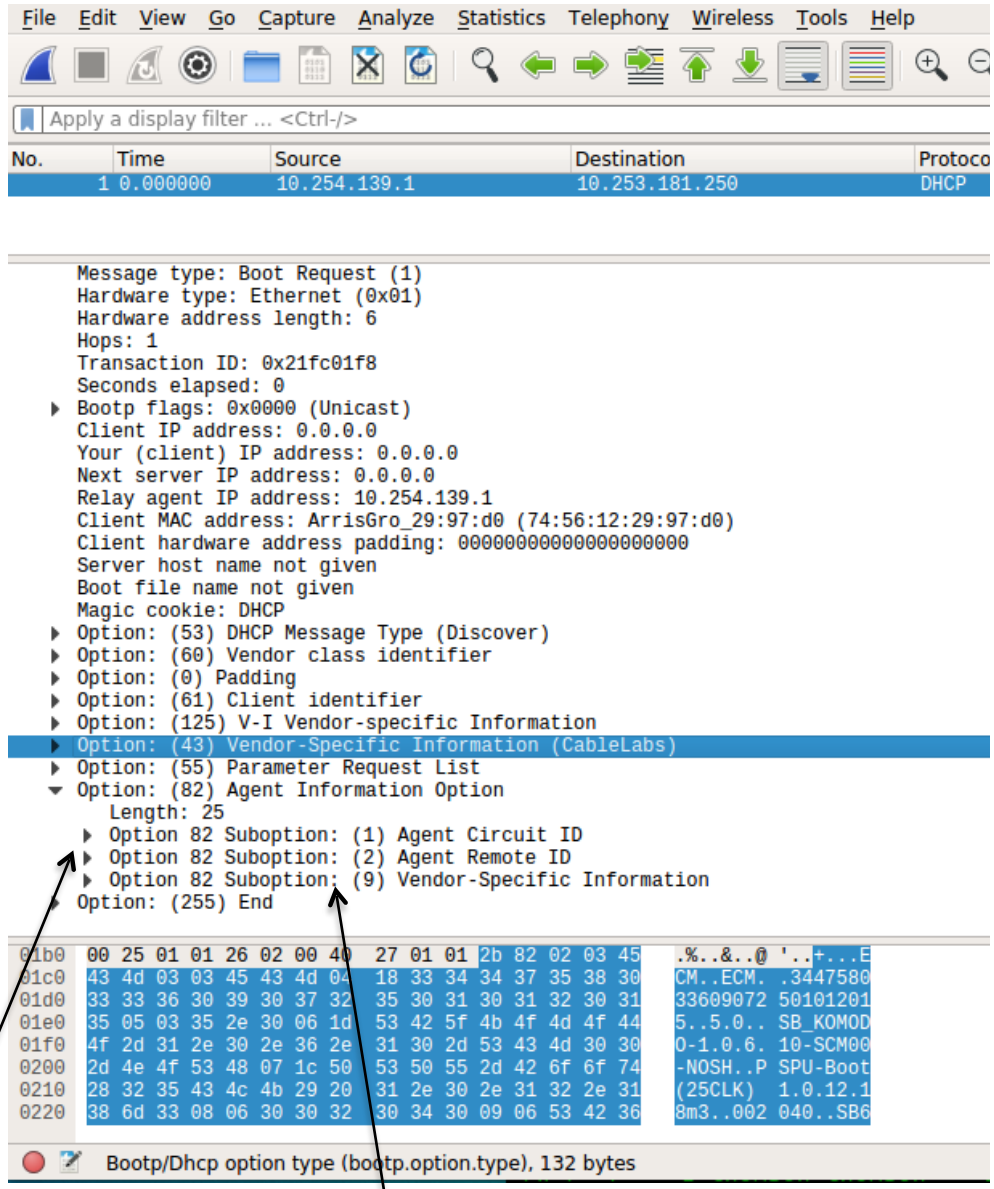
Photo by Angelo Failla, Facebook

Flex-id (1.2)

- Flexible Identifier

  How to identify hosts:
- **Open source**
- MAC, duid, circuit-id, client-id

- **Premium**
- Almost anything could be used (35 different expressions)
- Options (client, relay, vendor)
- Fixed fields
- Concat, substring
- Meta-data (interface name, src/dst IP, …)



concat(relay4[1].hex, relay4[2].hex)

REST API (1.2/1.3)

Overview:

- Command Channel (Unix socket)
- REST interface (http/https)
- JSON commands, JSON responses
- kea-shell provided (python 2.x, 3.x example)

Manipulate:

- Whole **config** (config-get/set/test/write)
- **Shared networks, subnets** (subnet4/6-list/add/get/del)
- **Host Reservations** (reservation-get/add/del)
- **Leases** (lease4/6-get/add/update/del/wipe)
- **Statistics** (statistic-get/reset/get-all)
- **Server** (list-commands, shutdown, version-get, build-report, leases-reclaim, etc.)

More to come in future releases

```
{
    "command": "subnet6-add",
    "arguments": {
        "subnet6": [ {
            "id": 234,
            "subnet": "2001:db8:1::/64",
            ...
        } ]
    }
}
```

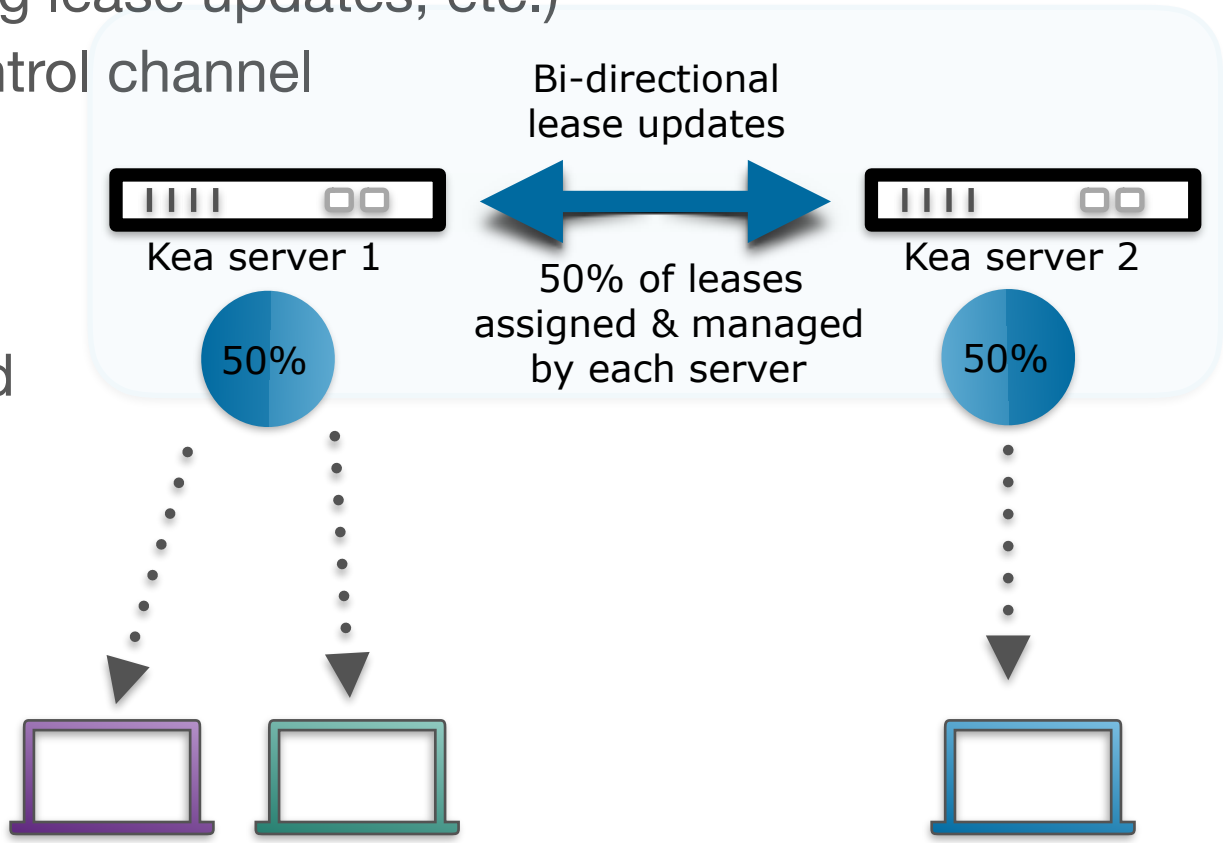
Command

```
{
    "result": 0,
    "text": "IPv6 subnet added",
    "arguments": {
        "subnet6": [
            {
                "id": 234,
                "subnet": "2001:db8:1::/64"
            }
        ]
    }
}
```

Response

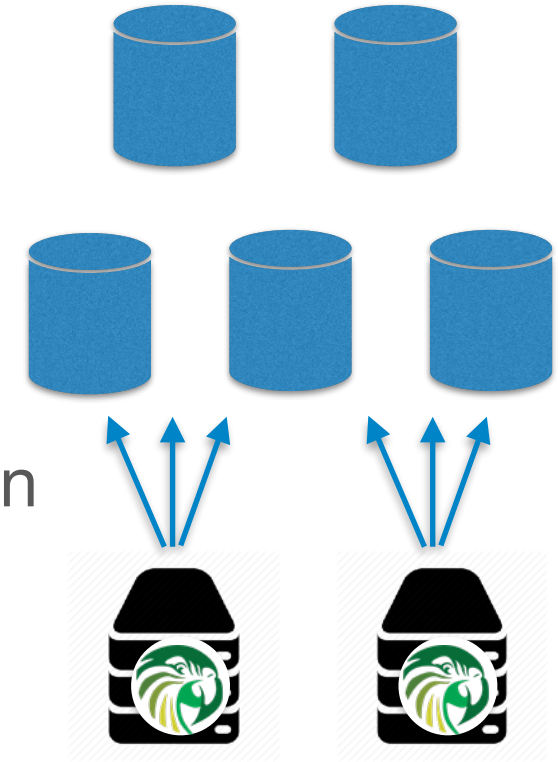
High Availability (1.4)

- **Load balancing** or **hot standby**
- RESTful API based
- Hook points (sending lease updates, etc.)
- Heartbeats over control channel
- Lease updates via lease_cmds hook
- Failure detection based on ‘secs’ field
- Auto-sync of lease database
- Backup server
- 50/50 LB split
- V4 and V6



Apache Cassandra (1.4)

- Distributed non-relational NoSQL database
- Massive scalability without a single point of failure
- Replication factor
- Can operate with at least one node surviving
- CQL
- Data denormalization



HA	NODES	FAILURES
	2	1

RF	NODES	FAILURES
1	1	0
2	3	1
3	5	2
4	7	3
	…	

RF = 2N + 1 can survive N failures

1.4 coming up

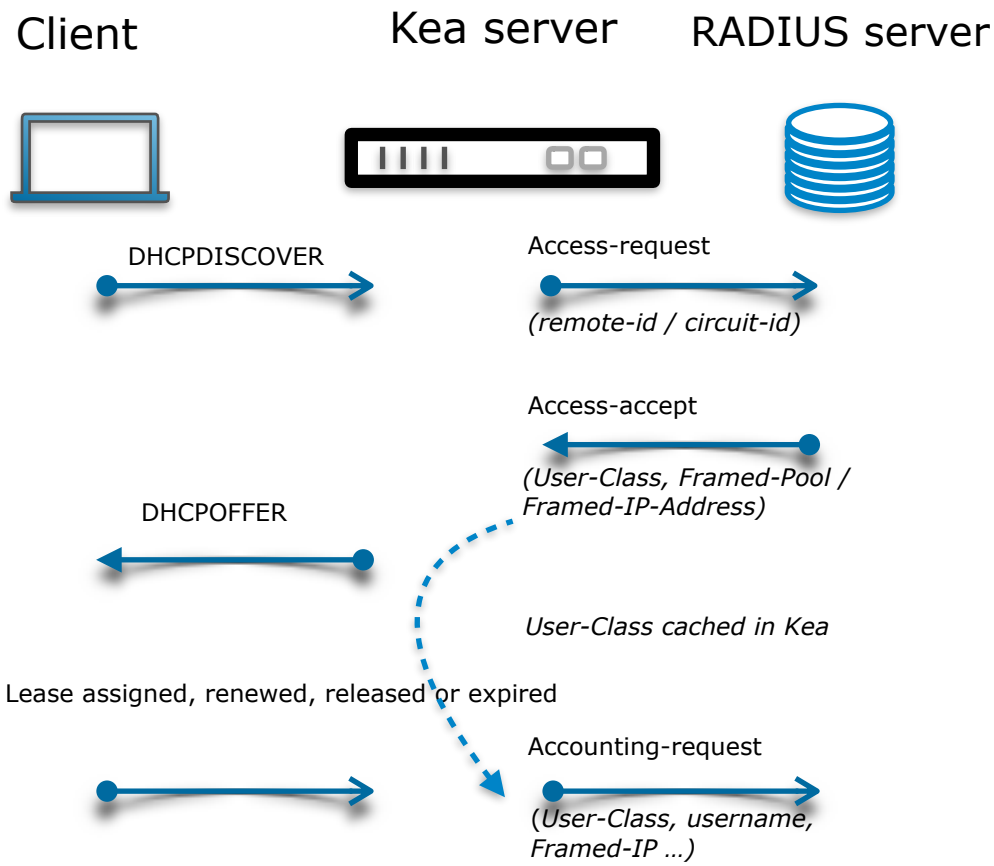
- 1.4.0 beta: **May 14th (18th)**, 1.4.0 final: June 15th
- Improved shared networks performance
- Improved classification
  - member(foo) && !member(bar) && (relay4[2].hex == ‘abcd’)
- Fixed statistics when run multiple instances with the same DB
- Many smaller bugfixes and improvements
  (100+ tickets closed and counting)

kea.isc.org/roadmap

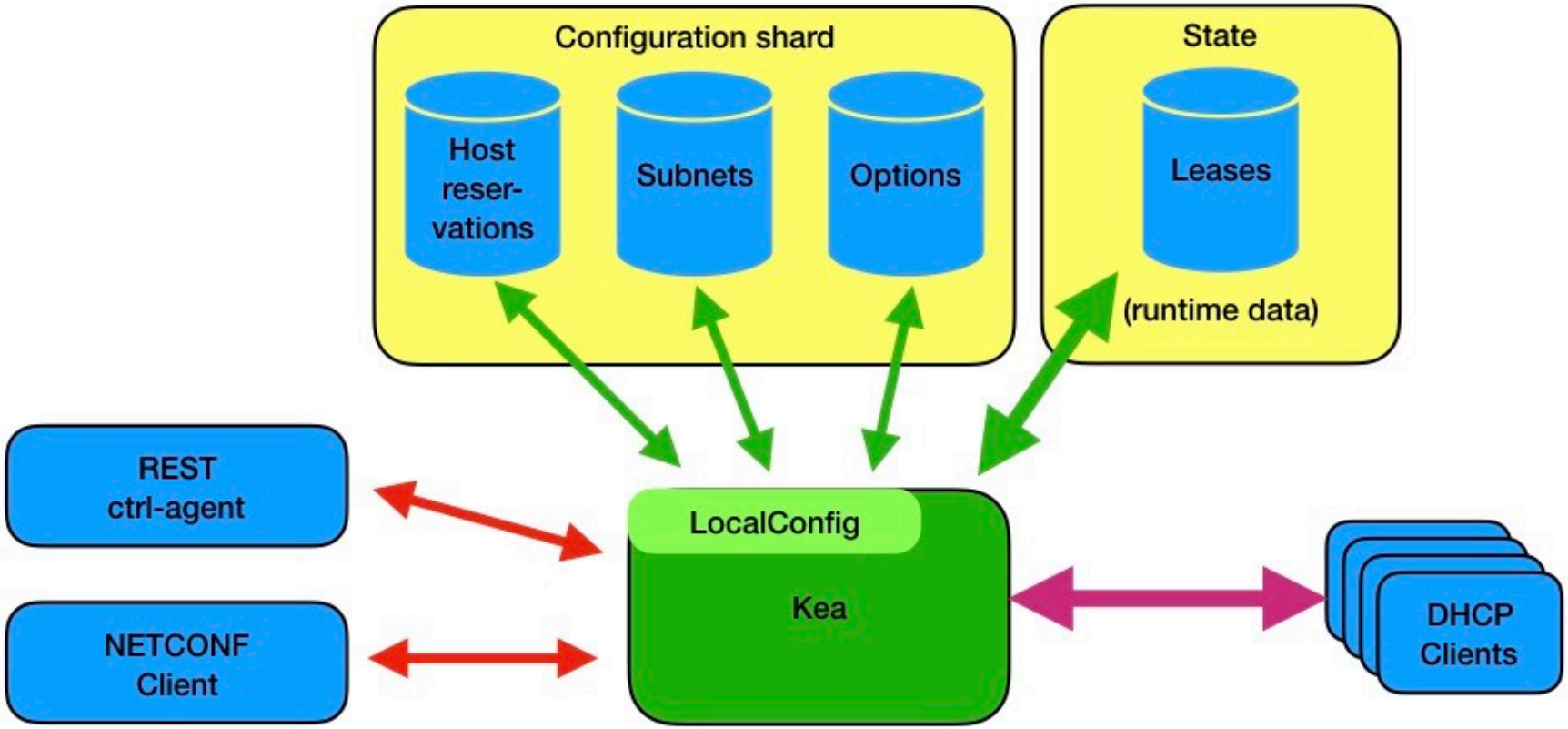
© 2018 ISC

RADIUS Integration (1.4)

- Authentication
  - Access control
  - Address Reservation
  - Class assignment
- Accounting
- Attributes customizable
- FreeRadius based
- DHCPv4 and DHCPv6



DB Configuration Storage (1.5)



kea.isc.org/wiki/CentralizedConfigNetconf

YANG/NETCONF (1.5)

IETF work: draft-ietf-dhc-dhcpv6-yang-06

```
<server xmlns="...">
  <server-config>
    <network-ranges>
      <option-set-id>1</option-set-id>
      <network-range>
        <network-range-id>1</network-range-id>
        <network-description />
        <network-prefix>2001:db8::/56</network-prefix>
        <option-set-id>2</option-set-id>
        <address-pools>
          <address-pool>
            <pool-id>1</pool-id>
            <start-address>2001:db8::1</start-address>
            <end-address>2001:db8::ffff</end-address>
            <renew-time>20</renew-time>
            <rebind-time>90</rebind-time>
            <valid-lifetime>150</valid-lifetime>
            <preferred-lifetime>120</preferred-lifetime>
          </address-pool>
        </address-pools>
      </network-range>
    </network-ranges>
  </server-config>
</server>
```

Configuration

```
container network-ranges {
    description "This model supports a hierarchy ...";
    list network-range {
      key network-range-id;
      leaf network-range-id {
          type uint32;
          mandatory true;
      }

      container address-pools {
      description "A container that describes the ...";
      list address-pool {
        leaf start-address {
            type inet:ipv6-address-no-zone;
            mandatory true;
            description "start address";
        }
        leaf end-address {
            type inet:ipv6-address-no-zone;
            mandatory true;
            description "end address";
        }
        leaf valid-lifetime {
            type yang:timeticks;
            mandatory true;
            description "valid liftime for IA";
        }
         ...
      }
```

YANG model

TreeView for YANG model

Module: ietf-dhcpv6-server@2018-03-04, **Namespace:** urn:ietf:params:xml:ns:yang:ietf-dhcpv6-server, **Prefix:** dhcpv6-server
Impact Analysis for ietf-dhcpv6-server@2018-03-04

Element [+] Expand All [-] Collapse All	S	Type	F	C	S	Path
ietf-dhcpv6-server	m	module				
server	c	container	c	F	c	/dhcpv6-server:server
server-config	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config
serv-attributes	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:serv-attributes
option-sets	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:option-sets
network-ranges	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges
option-set-id	le	leafref	c	?	c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
network-range	lis	list	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
network-range-id	le	uint32	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
network-description	le	string	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
network-prefix	le	inet:ipv6-prefix	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
option-set-id	le	leafref	c	?	c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
address-pools	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
pd-pools	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
host-reservations	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:network-ranges/dhcpv6-serve
relay-opaque-paras	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:relay-opaque-paras
rsoo-enabled-options	c	container	c		c	/dhcpv6-server:server/dhcpv6-server:server-config/dhcpv6-server:rsoo-enabled-options
server-state	c	container	r		c	/dhcpv6-server:server/dhcpv6-server:server-state
dhcpv6-server:notifs						

Useful links

- **Kea project homepage**: http://kea.isc.org
- **Documentation**: http://kea.isc.org/docs/
  - User's Guide - 100+ pages of guidance with examples for users, REST API documentation, and user documentation for premium hooks (easy to see if you would benefit from purchasing them)
  - List of all log messages - with an explanation what happened and why, a nod towards the mainframe era
  - Developer's Guide - for developers and contributors, explains the internals, also includes Hooks interface API
- **Kea business page**: http://isc.org/kea
  - High level overview, premium hooks white papers, ISC DHCP vs Kea comparison, support links, 24/7 support available
- **The source code**: http://github.com/isc-projects/kea
  - Source code for premium hooks is also provided to purchasers

Q&A

Questions?

Suggestions?

Tomatoes?

© 2018 ISC