# Kea – modern DHCP server

## Open Source and Sustainability

Tomek Mrugalski
tomasz@isc.org

# What is Kea?

## And why you may want to use it

# If you never heard about Kea…

- Modern DHCPv4 and DHCPv6 server (1.0 in Dec 2015)
- Performance (1000s leases/sec)
- Scalable (millions of devices)
- No restarts after config changes
- Databases (CSV, MySQL, PostgreSQL, Cassandra)
- Hooks (3rd party libraries)
- REST management API
- Linux, BSDs, MacOS, …
- Open source (MPL2)
- 1.2.0 just released (28 April 2017)

# Let’s compare!

| | ISC DHCP | ISC Kea |
|---|---|---|
| Started | prehistory (1995) | Recent (2011) |
| Code | Not adding anything big | Active development with tons of new features |
| Code repository | Internal, tarball published | github |
| Bug database | Internal, mail external | public trac |
| Testing | ~30 unit-tests | 4000+ unit-tests<br>Memory leak tests (valgrind)<br>700+ system tests<br>Fuzz testing |
| Docs | Man pages | User’s Guide (100+ pages)<br>Developer’s Guide |
| Logs | Fixed log message | Every possible log entry is documented and described |
| IPv6 readiness | IPv4 originally, IPv6 added later | IPv4 optional |

# Why migrate from ISC DHCP?

| | ISC DHCP | ISC Kea |
|---|---|---|
| Performance | Ok (with ramdisk tricks) | Great<br>(many 1000s leases/sec) |
| Management | OMAPI (custom C interface) | **JSON over REST API/http**,<br>JSON over Unix socket |
| Extensibility | Shell scripts (out only),<br>configuration language | **JSON** everywhere,<br>**Hooks** (C++), stable API |
| Configuration | Custom complex syntax<br>(almost programming<br>language) | **JSON with optional DB storage**<br>for some elements (more to come) |
| Leases<br>information | Custom | **CSV, MySQL, PgSQL, Cassandra** |
| Hosts information | Custom config | **JSON, MySQL, PgSQL** |

# Cool features :: DB


- Leases, host reservations in DB
  - CSV
  - MySQL or PostgreSQL
  - Cassandra*


- SQL data can be modified any time
- All changes applied instantly (no restart)


- Can fiddle with the DB directly or
- Use host commands (1.2) or subnets (1.3)

# Cool features :: REST

- Command Channel (Unix socket, since 0.9.2)
- REST interface (http, since 1.2.0)
- JSON commands, JSON responses
  - kea-shell provided (python 2.x, 3.x example)
  - Trivial to use from any JSON/http capable env
- Commands:
  - config-get, config-set, config-test, config-write
  - reservation-get, reservation-add, reservation-del*
  - statistic-get, statistic-reset, statistic-get-all, statistic-reset-all, …
  - leases-reclaim, list-commands, shutdown, version-get, build-report
- More to come every release

# Kea Roadmap

- REST interface
- Rewritten configuration handling
- Commands (config-set/get/test/write)

**1.2**
**(Apr 2017)**

**Host Commands**
**Flexible Identifier**

- Shared subnets
- Security for REST interface
- Lease commands
- …

**1.3**
**(Oct 2017)**

**Subnet Commands**

- Multi-proc support
- Better High Availability/Redundancy
- DB improvements
- YOUR FEATURE HERE

TBD

**1.4**
**(spring 2018)**

Open source

Premium

# Cool features :: Hooks

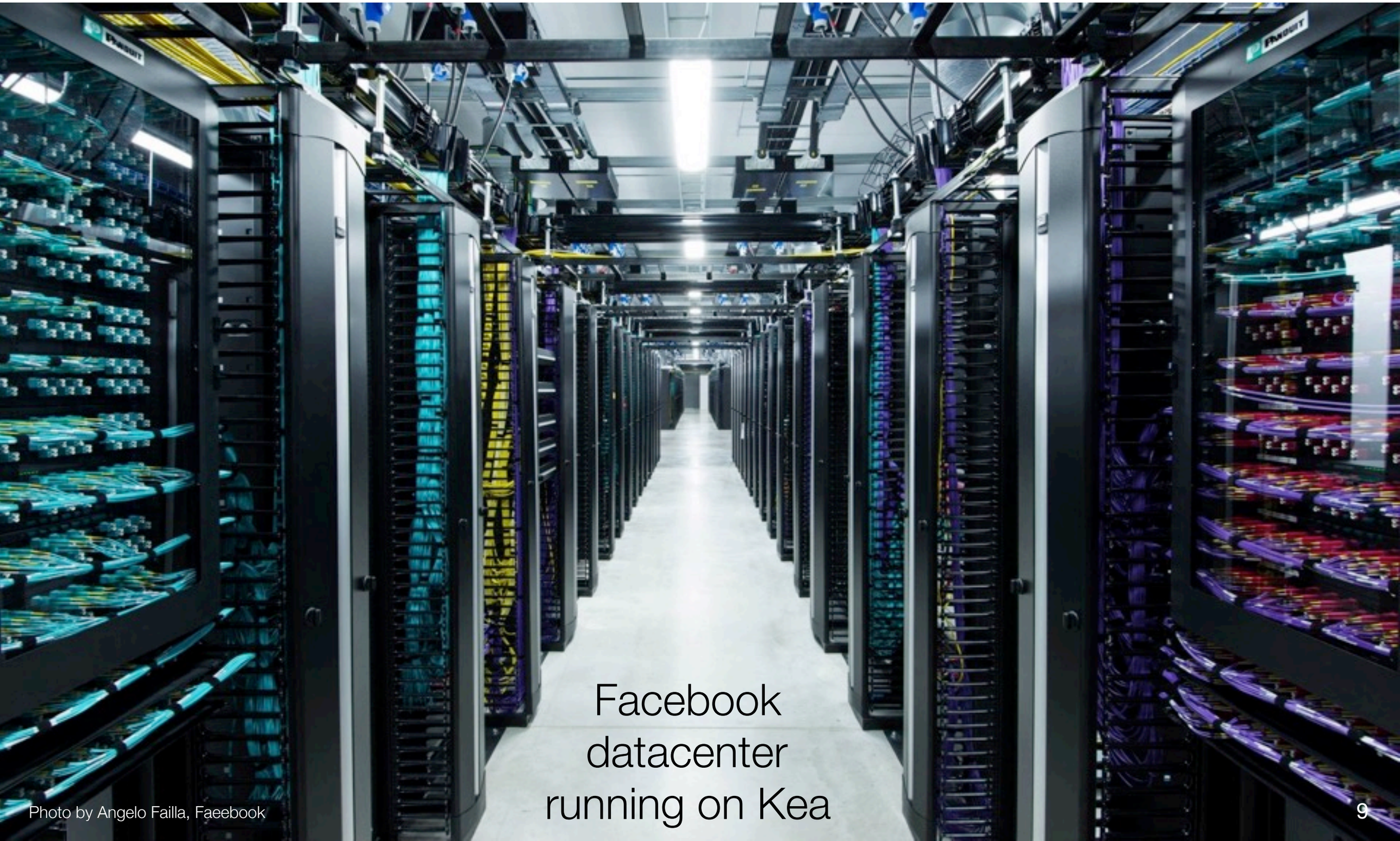



Photo by Angelo Failla, Faeebook

# Open Source and Sustainability

## How to properly fund OS?

# Commercial quality software

- A small team (2 full time, with 2 more contributing) of experienced engineers
- A real, independent QA
  - 4000+ unit-tests, 700+ system tests
  - Run on ~20 systems
  - Valgrind, Coverity scan, other static analyzers
- Proper designs
  - Written Requirements, Designs, Implementation, Testing
- Very well documented
  - User's Guide (100+ pages, ~40 example configs)
  - Developer's Guide (code is well commented, all params documented)

# Funding so far

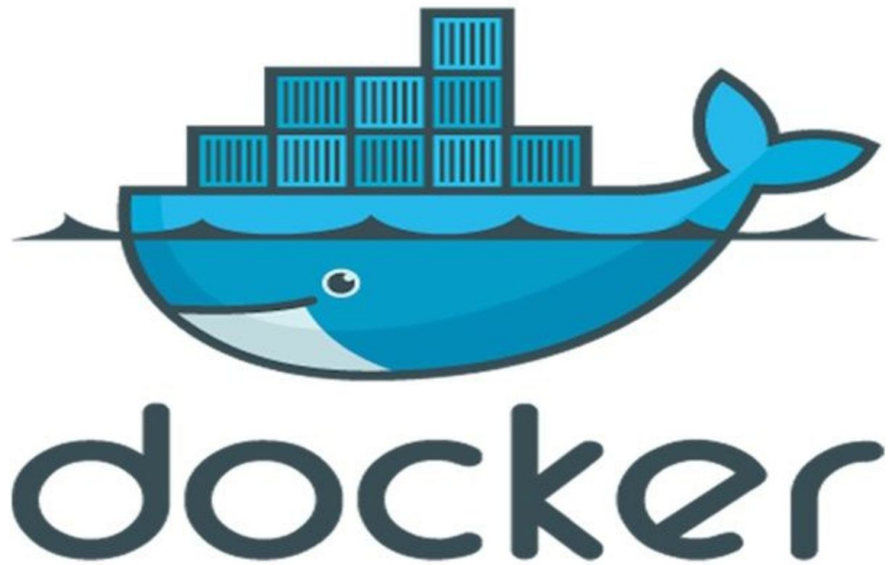
- In development since 2011
- Had several custom development contracts
- Two sponsors (Comcast and Mozilla, thanks!)
- Very few support customers
- Sporadic personal contributions (thanks!)

But…

Most of the work was
internally funded by ISC

# Funding Idea #1: Kea 1.2.0 Docker

- Docker image with Kea + MySQL pre-configured
- Easy to deploy
- An experiment



# Funding Idea #2 : Premium features

- Kea is and will remain open source (MPL2, 478KLOC)
  - Provides support for hook libraries (~Apache module)
  - API is open (3rd party hook libs appearing now)

- Premium (EULA, 6.8KLOC)
  - Additional extra features
  - targeted for large deployments
  - A way to convince people to sign support contract
  - Yes, support contract = $

isc.org/blogs/funding-kea/

# Existing & Planned Hooks

1.2

- User_chk – example access control (open source)
- Forensic Logging – detailed audit trail for legal purposes
- Flexible Identifier – identify hosts by expression, e.g. concat(relay4[2].hex, relay4[6].hex)
- Host Commands – query, add and delete host reservations using REST interface

1.3

- Subnet management (add, get, update, delete)
- Extra lease commands (add, get, update, delete)

Open source  Premium

# Hook Example

## Flexible Identifier

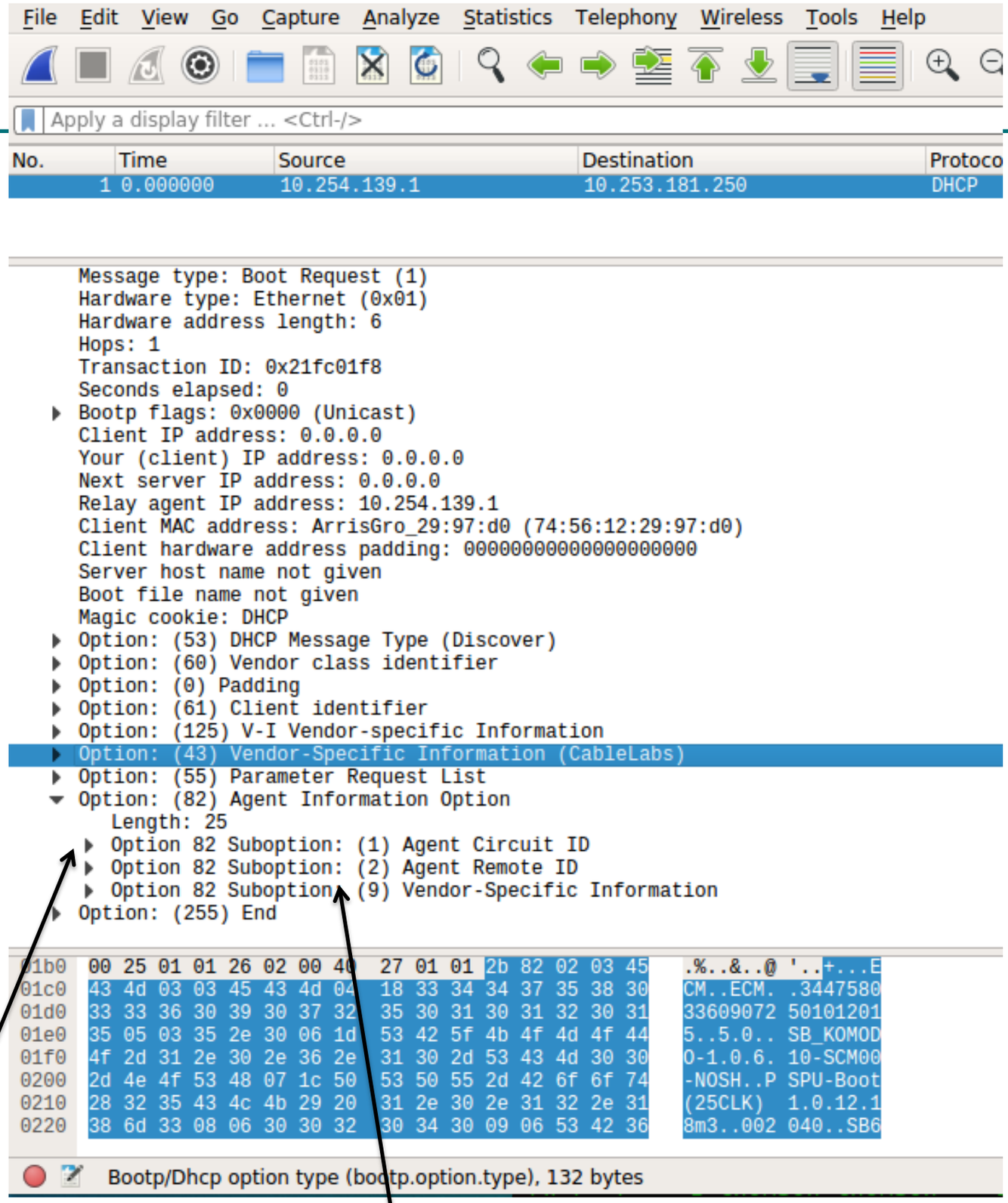
How to identify hosts:

**Open source**

- MAC, duid, circuit-id, client-id

**Premium**

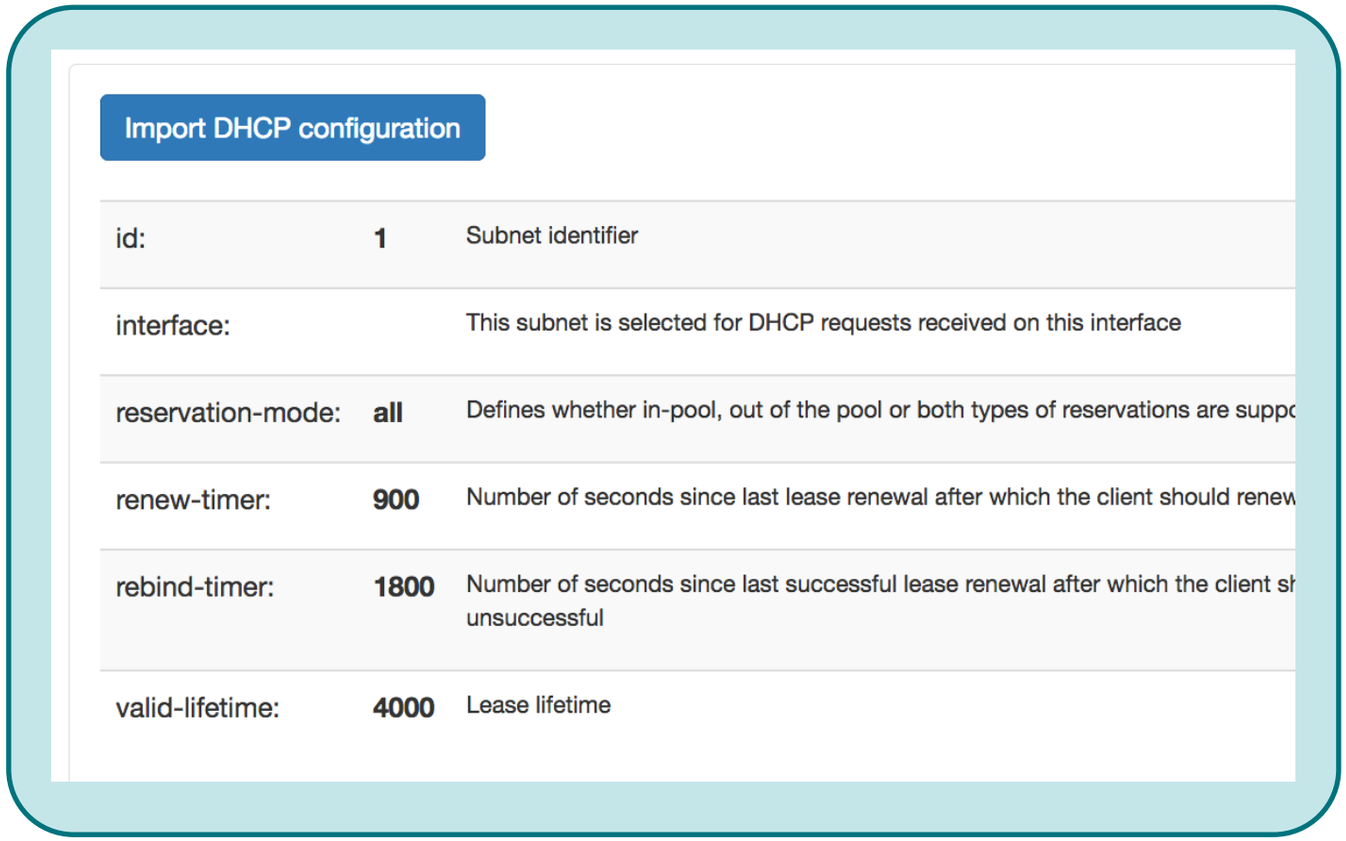
- Almost anything could be used (35 different expressions)
- Options (client, relay, vendor)
- Fixed fields
- Concat, substring
- Meta-data (interface name, src/dst IP, …)

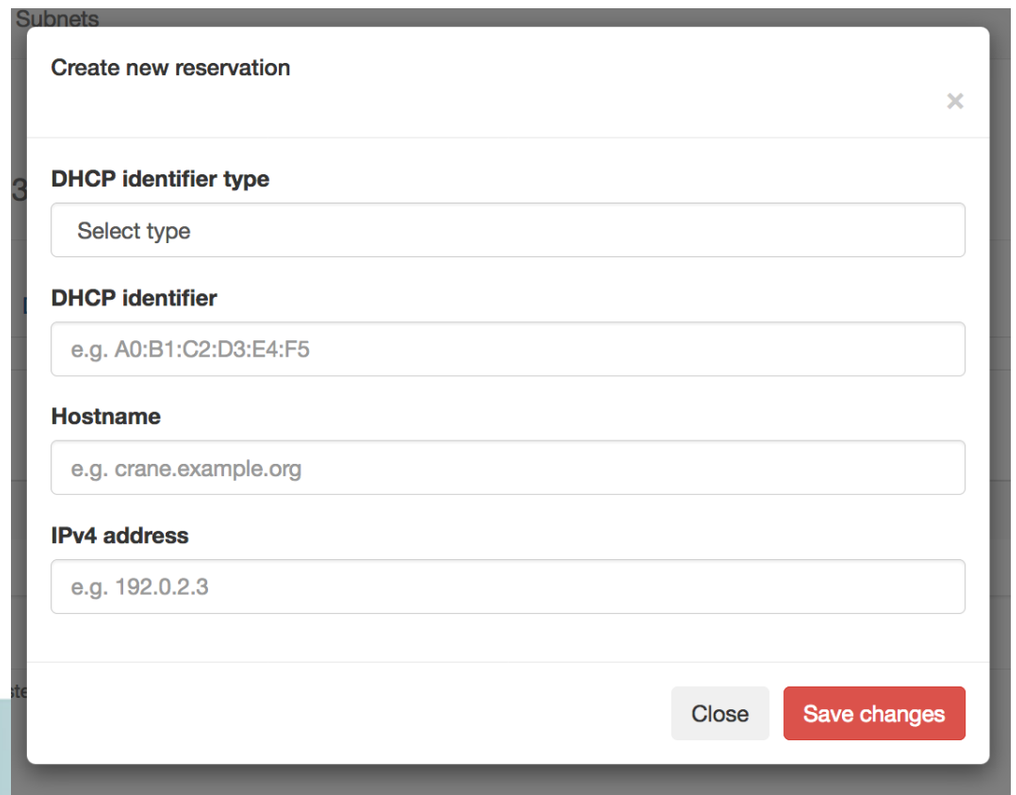



concat(relay4[1].hex, relay4[2].hex)

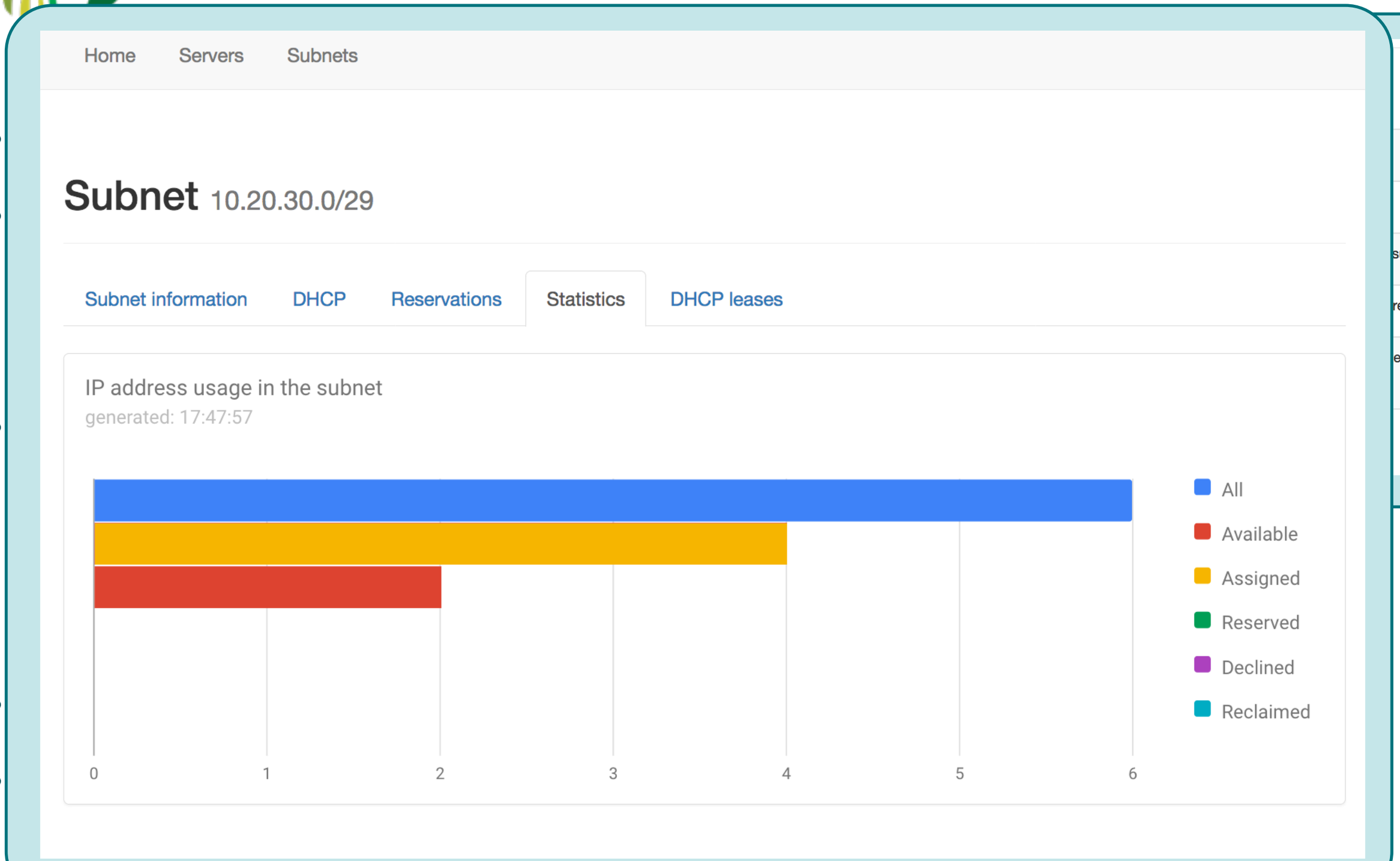
# Funding Idea #3 : Kittiwake GUI

- REST interface
- Focusing on typical ops first:
  - Monitor pool utilization
  - Add/delete host reservations
- Distribution/Licensing TBD
  - Freemium?
  - Paid app?
  - Extra benefit for customers?
- 1.0 planned in Autumn 2017
- Get in touch! We'll be looking for beta testers/your requirements



# Funding Idea #3 : Kittiwake GUI



# Funding Idea #4: ISC DHCP to Kea migration

- Migration tool is in development
- ISC DHCP config is complex (80% cases)
- Trials will start soon
- Interested?
  - We’re looking for configuration samples
  - Talk to us
- Revenue model TBD

Your thoughts
on those ideas?

Questions?

Suggestions?