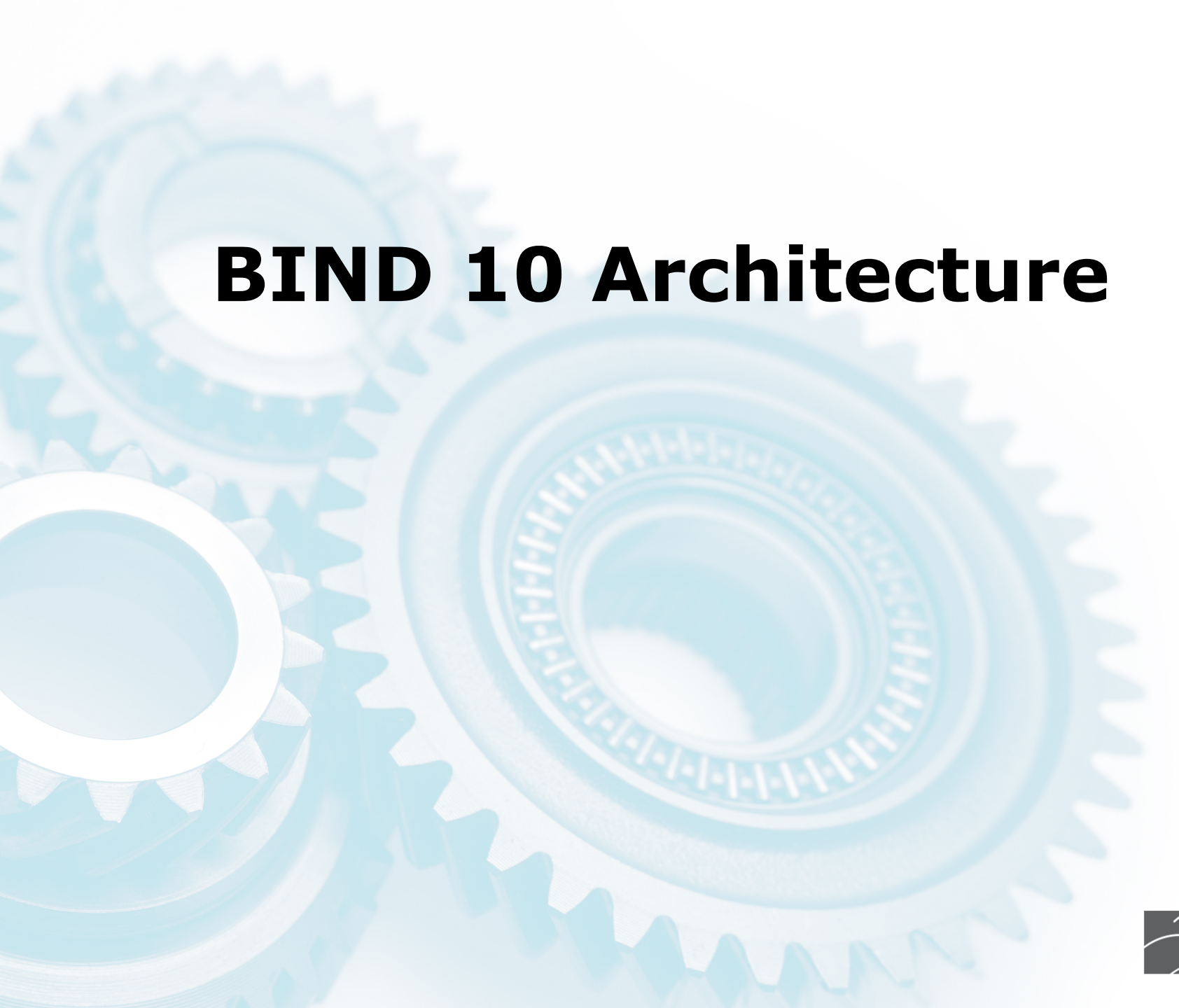
BIND 10 Architecture



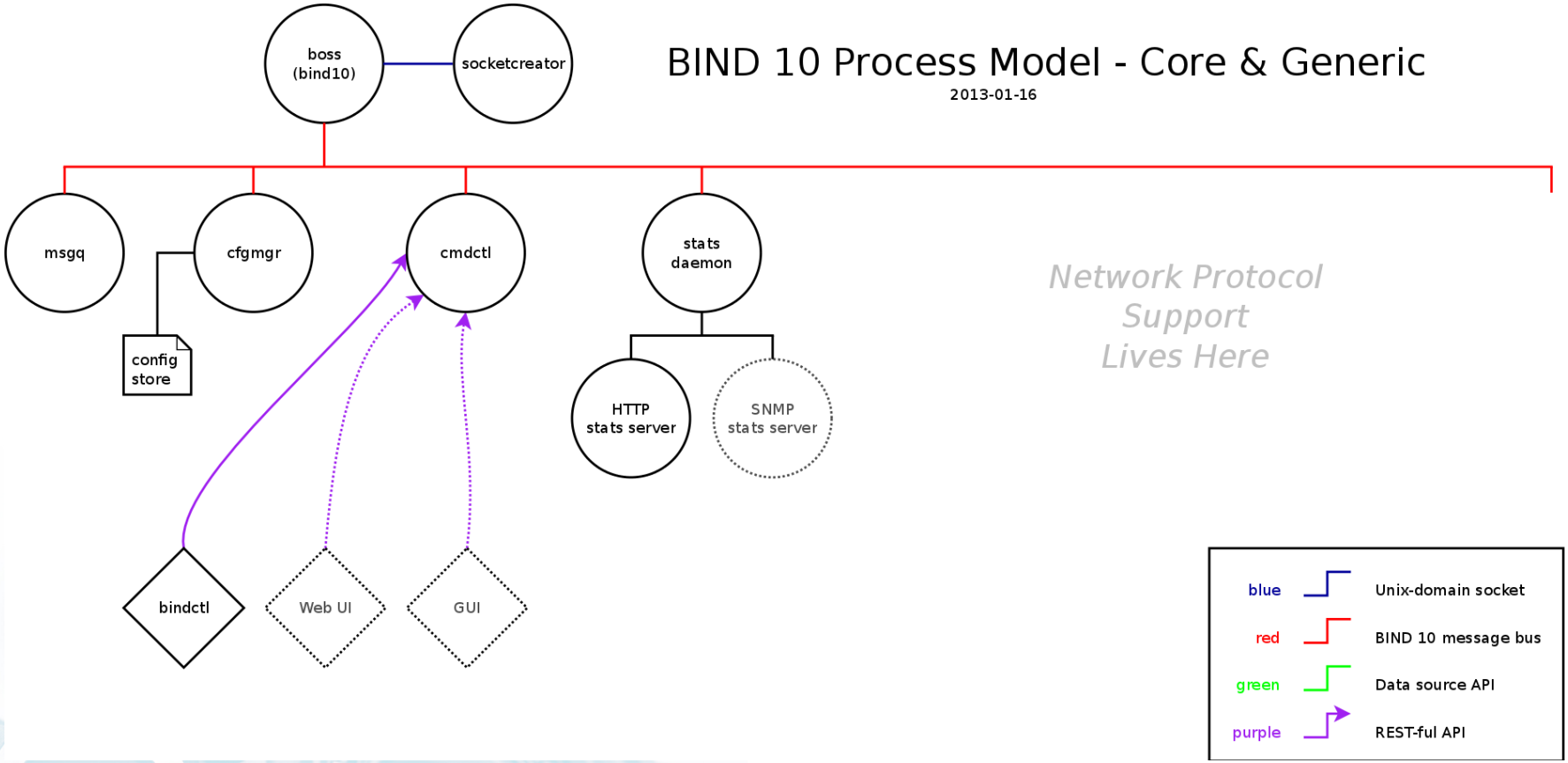
Part the First: Generic BIND 10

Goals that Affect Architecture I

- Customization "out-of-the-box"
  - authoritative-only, recursive-only
  - slave-only, master-only
  - DHCPv4-only, DHCPv6-only
  - enable/disable dynamic DNS
  - support favorite SQL backend
- Customization via code changes
  - non-ISC modules, or modifications
  - bespoke or in-house development

Goals that Affect Architecture II

- Scalability
  - BIND 8/ISC DHCP: single core
  - BIND 9: multiple cores (4-6 or so)
  - BIND 10: 10's or 100's of cores, multiple machines (clustered)
- Robustness
  - Reduce serious software bugs
  - Minimize impact of bugs
  - Reduce “fate sharing”



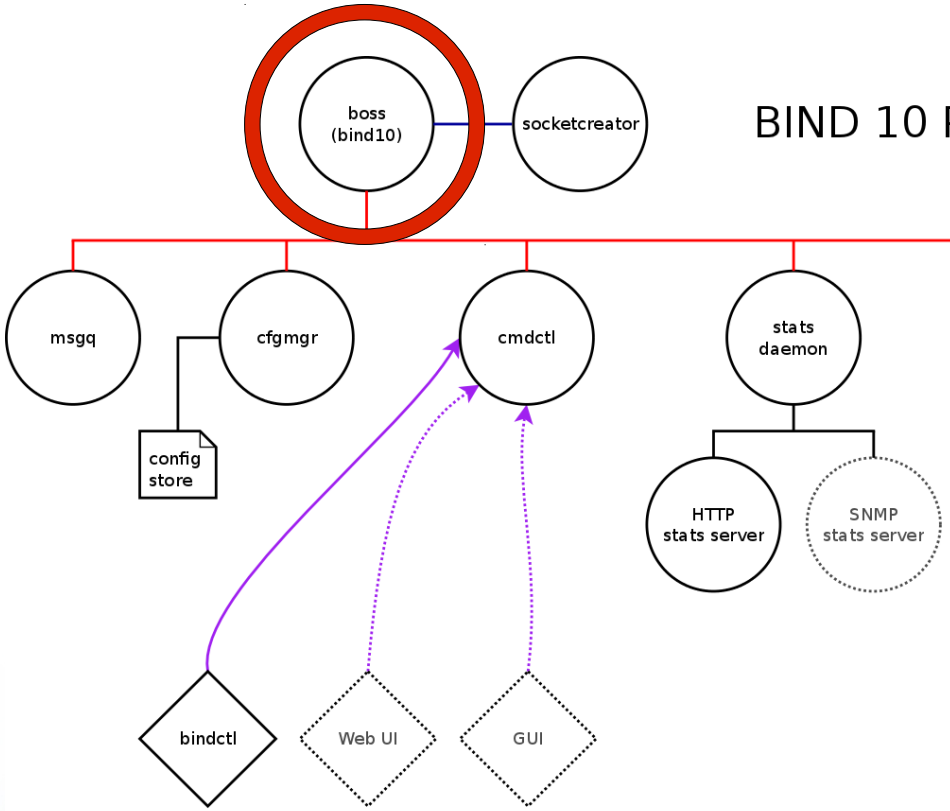
https://bind10.isc.org/wiki/DesignDiagrams

BIND 10

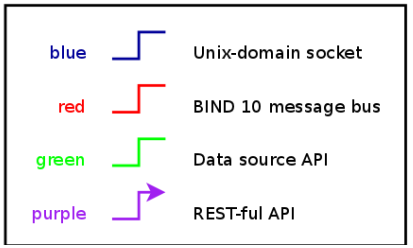
- Core
  - boss
  - msgq
  - cfgmgr

BIND 10 Process Model - Core & Generic

2013-01-16



Network Protocol Support Lives Here



Master^WBoss of BIND

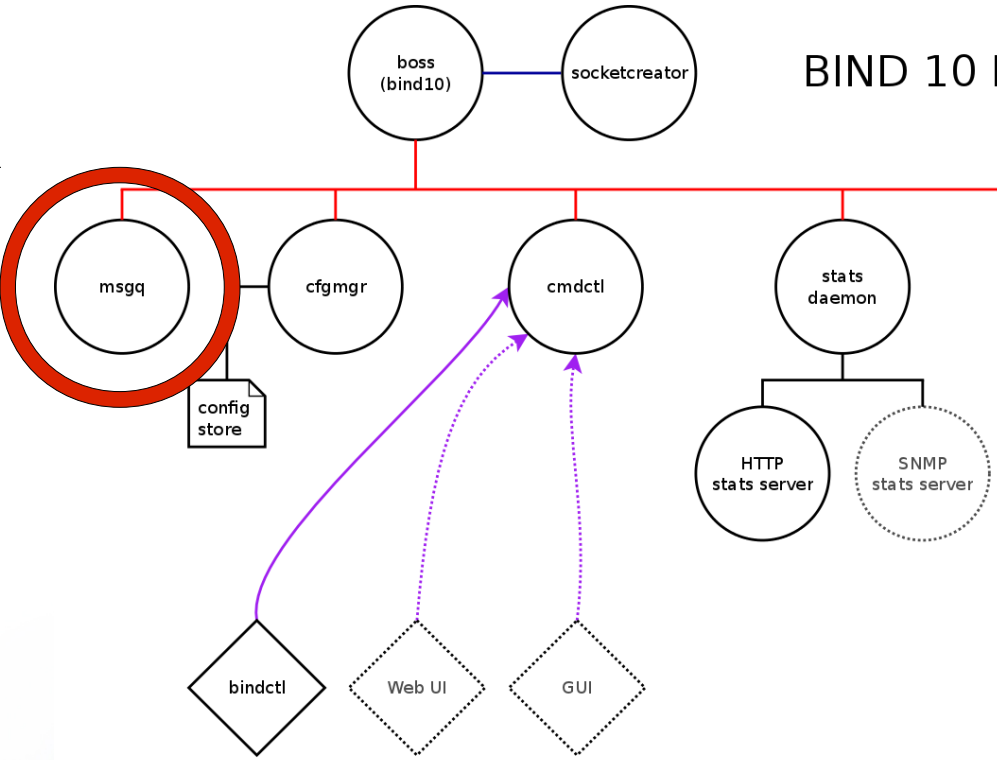
- Handles startup, shutdown
- Restarts processes that die
- Written in Python

An Aside: BIND 10 Languages

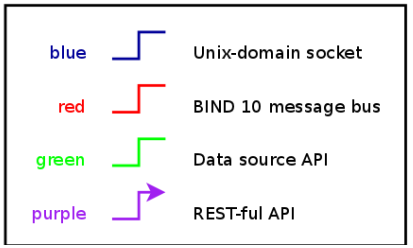
- C++ for performance critical parts
  - Modern compiled language
  - Widely used
- Python for... everything else
  - Modern scripting language
  - Widely used
  - Chose Python 3.x
  - Best. Language. Ever.

BIND 10 Process Model - Core & Generic

2013-01-16



Network Protocol Support Lives Here

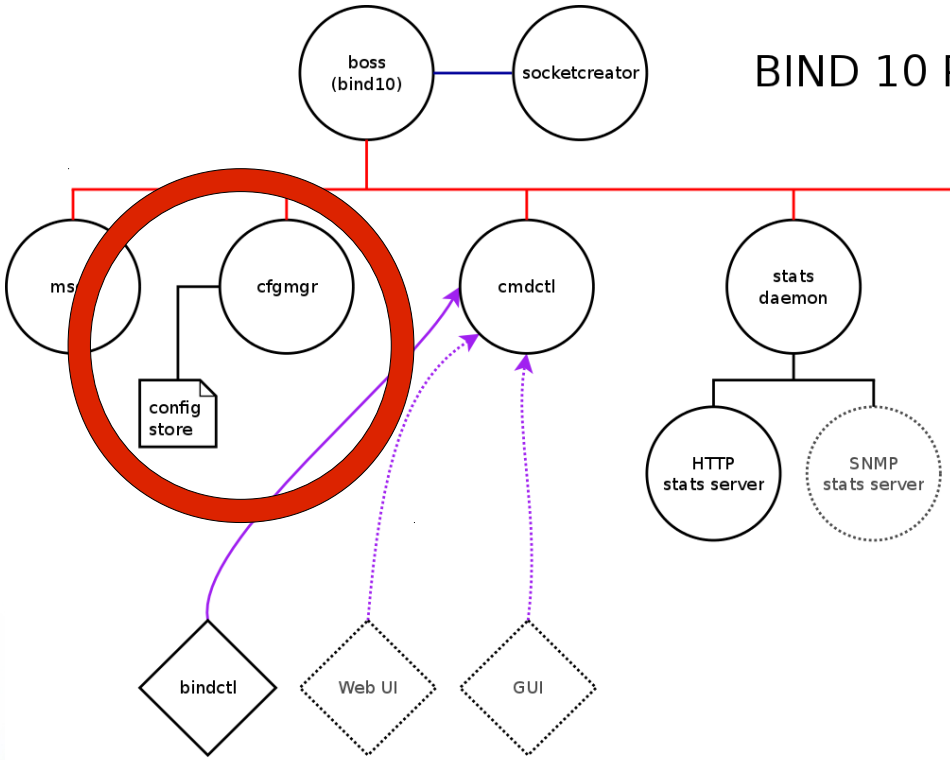


msgq

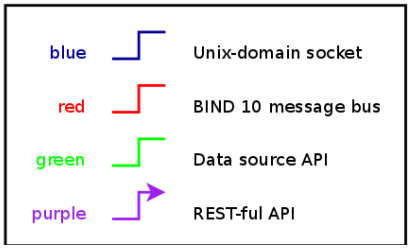
- Inter-process message bus
- Needed for extensibility
- Like d-bus, also for inter-machine
- Internal message format: JSON
- Unix domain socket connections
- No internal security

BIND 10 Process Model - Core & Generic

2013-01-16



Network Protocol Support Lives Here



cfgmgr

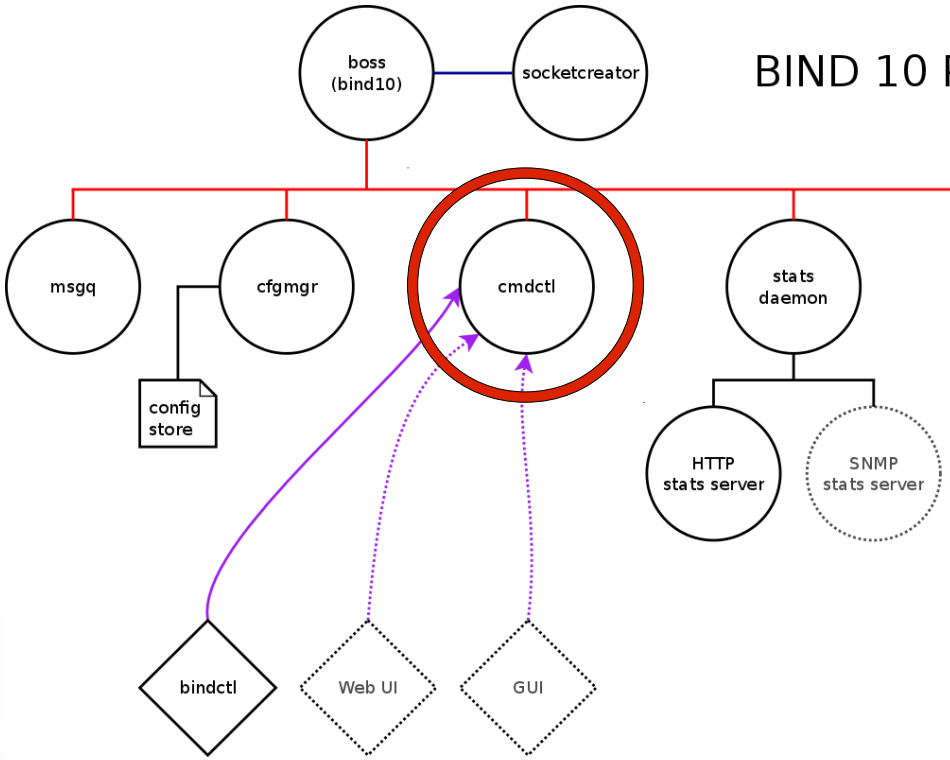
- Configuration manager
- Never need to restart BIND 10
- Flexible, extensible configuration
- Not traditional Unix configuration
  - Changes are immediate & persistent
  - More like a router or an application

BIND 10

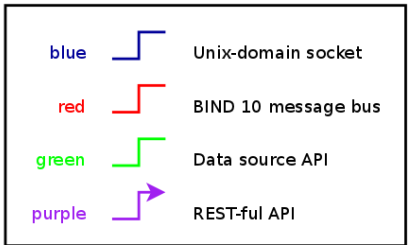
- Core
  - boss
  - msgq
  - cfgmgr
- Optional
  - cmdctl
  - auth
  - xfrin/xfrout
  - zonemgr
  - stats
  - ddns/dhcpdns
  - dhcp4/6

BIND 10 Process Model - Core & Generic

2013-01-16



Network Protocol Support Lives Here

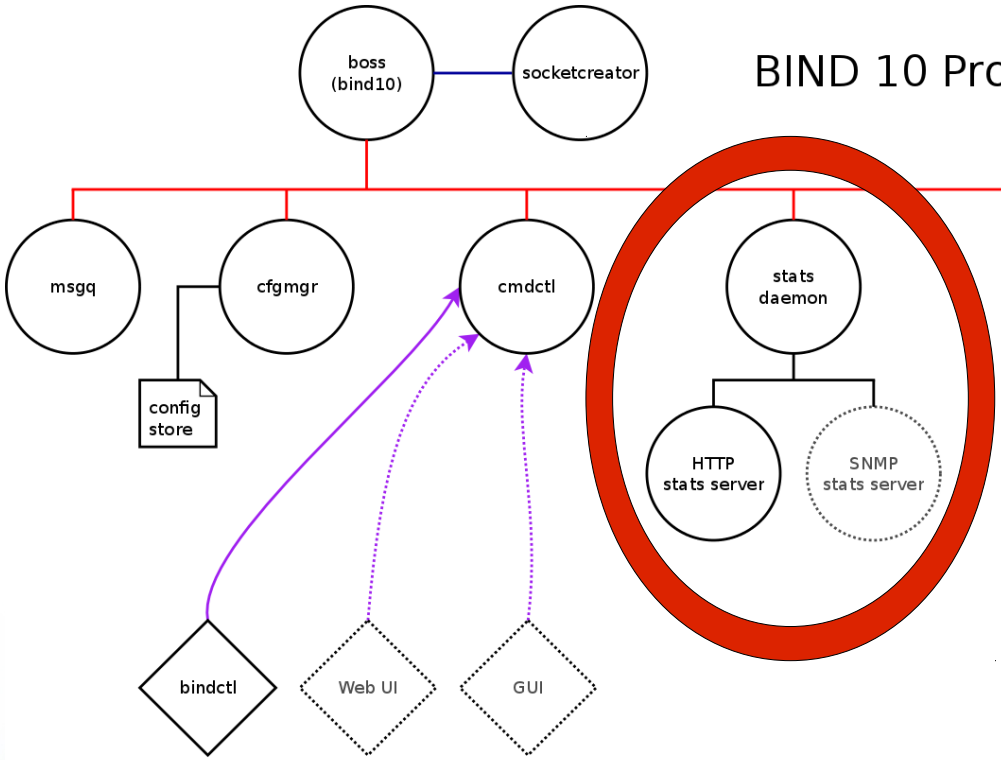


cmdctl

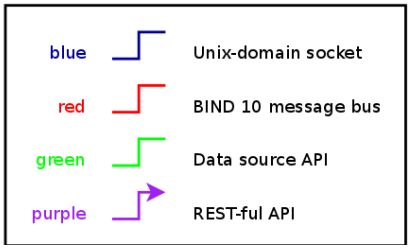
- Used to control the server
- Authenticates users
- Interacts with cfgmgr
  - Gets per-module options
- Interacts with modules
  - Commands like “refresh zone”
- Current client: bindctl (CLI)
- Future clients: web, GUI, wizards

BIND 10 Process Model - Core & Generic

2013-01-16

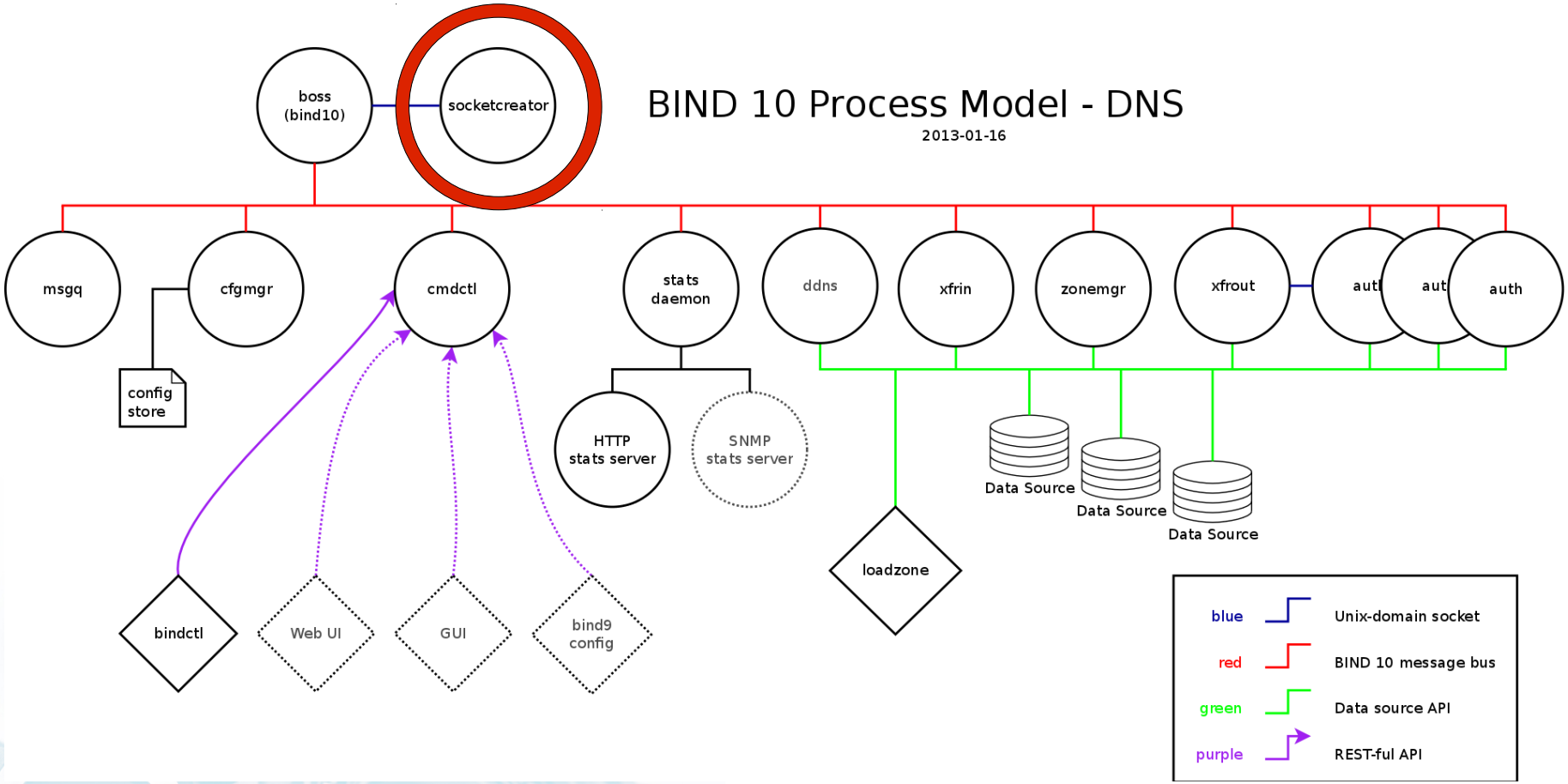


Network Protocol Support Lives Here



Statistics

- Modules report stats
- Collected by stats daemon
- Presented in various ways:
  - Via bindctl
  - XML over HTTP (BIND 9 style)
  - SNMP (planned)


BIND 10 Process Model - DNS
2013-01-16
boss
(bind10)
socketcreator
msgq
cfgmgr
config
store
cmdctl
stats
daemon
HTTP
stats server
SNMP
stats server
ddns
xfrin
zonemgr
xfrout
auth
auth
auth
Data Source
Data Source
Data Source
loadzone
bindctl
Web UI
GUI
bind9
config
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

Privileged Socket Creator

- Ports < 1024 restricted to root
  - DNS runs on port 53
- We want to drop permissions ASAP
- We want port 53 at any time!
  - Administrator may reconfigure
- Solution: Privileged Socket Creator
  - Small, single purpose C++ program
  - Uses file descriptor trick to send sockets around

Supporting Library: Logging

- Messages have unique identifiers
  - Only used 1 place in code
  - Have short and full explanations
- log4cplus
  - Like log4j Java library
  - Can turn logging off per module
- Message manual

http://bind10.isc.org/docs/bind10-messages.htm

Extending BIND 10: Hooks

- Allows targeted behavior changes
  - E.g. Modify reply packets
  - E.g. Invoke back-end processes
- Similar to plug-in or extensions
- Loadable at run-time
  - C++ initially, then Python
  - May extend to other languages
- Provide API for developers

Part the Second: BIND 10 DNS

Supporting Library: Data Sources

- Idea stolen from PowerDNS
- Back-end for authoritative DNS
- Currently SQLite or in-memory
- Plans:
  - MySQL, PostgreSQL
  - Berkeley DB
- Used by auth, xfrin, xfrout, loadzone, ddns

Data Source: SQLite

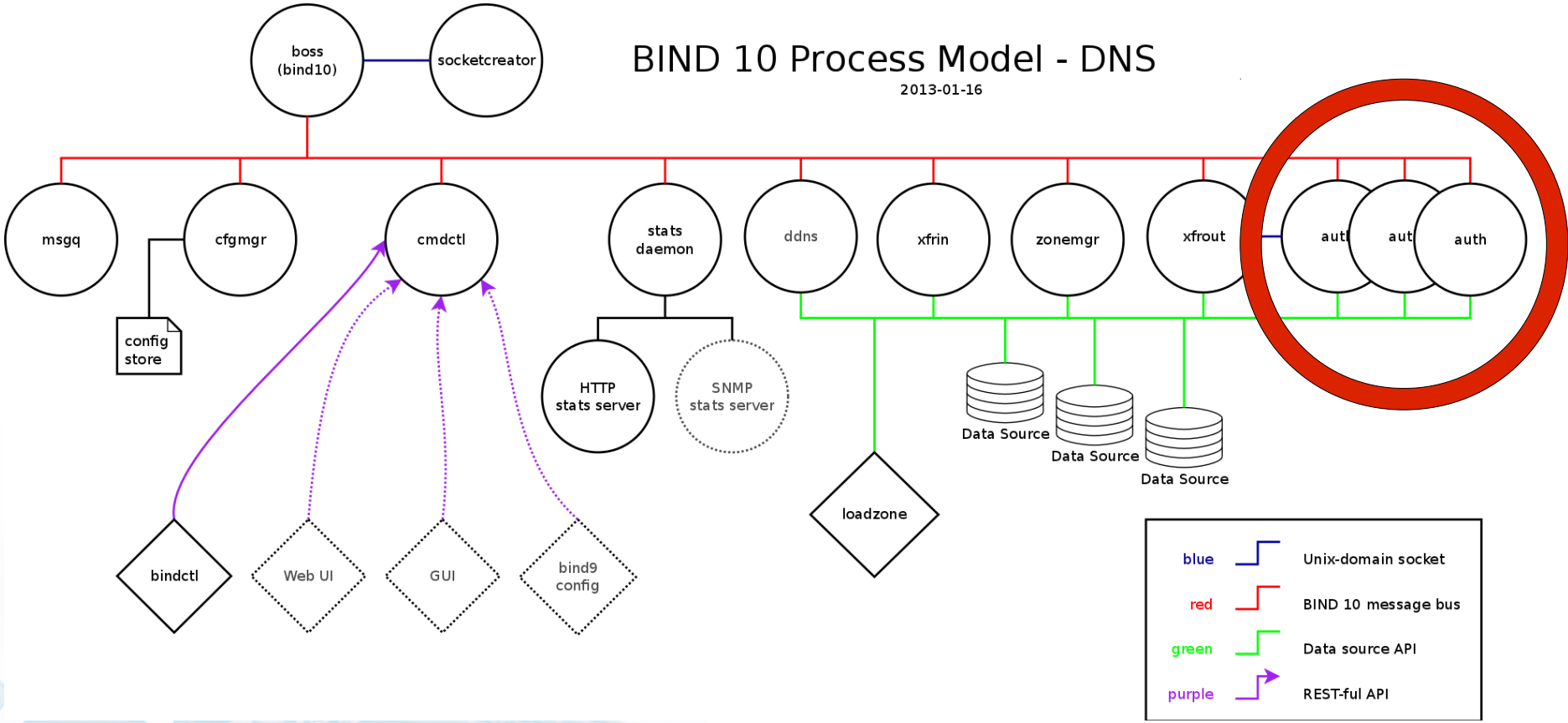
- Simple for administrators
  - "built-in", only file name needed
- Performance: reasonable
  - Much slower than in-memory
  - Good single-access performance
  - Collapses under heavy writing
- Pre-defined schema
- "instant on" for zones

Data Source: In-memory

- Based on BIND 9 red/black trees
- Performance reasonable
  - Similar to BIND 9
- Memory footprint good
  - Much smaller than BIND 9
  - Currently no shared memory
- Zones have to be loaded
  - Basically like BIND 9

Supporting Library: libdns++

- Low-level DNS messages (packets)
- C++ implementation
- Python wrapper

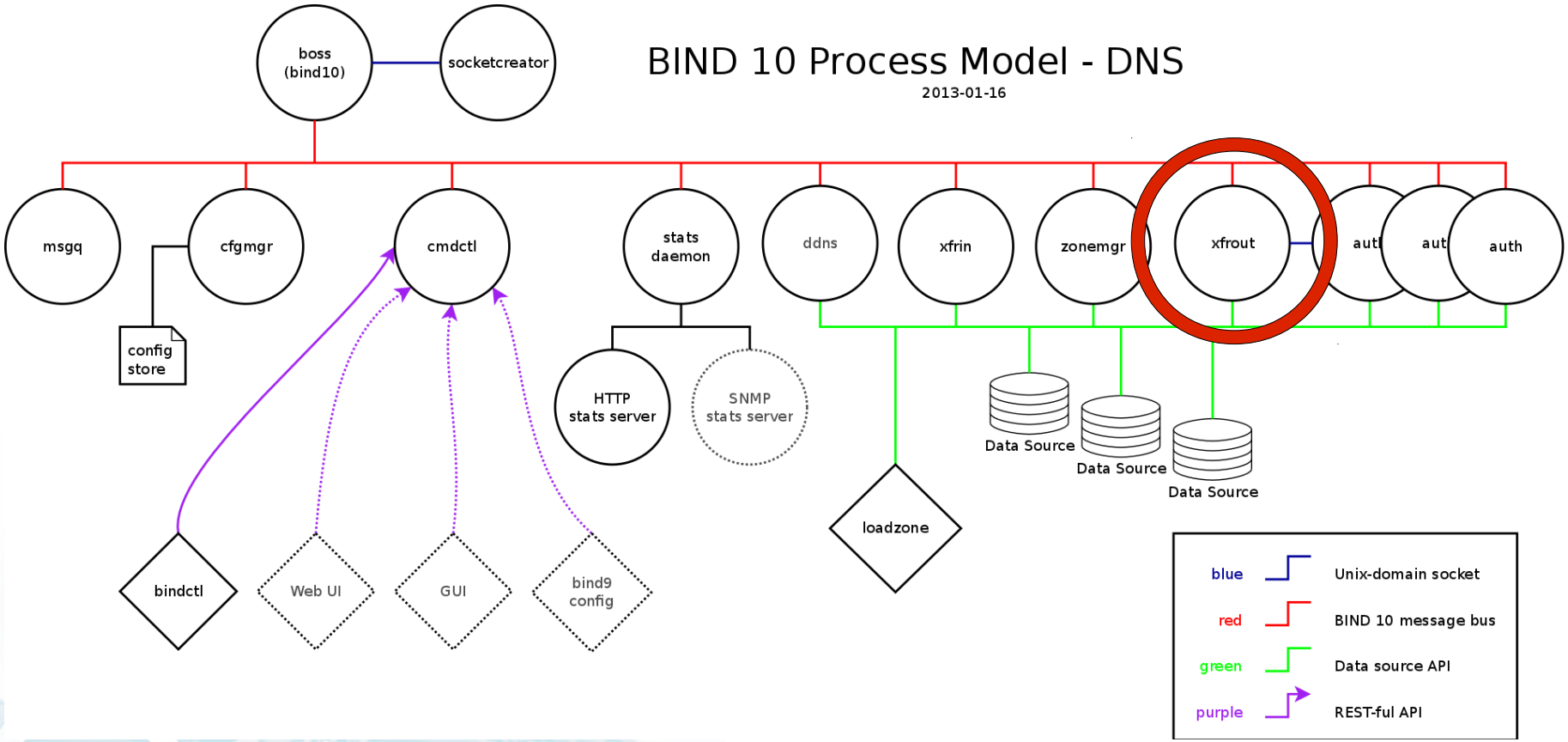

BIND 10 Process Model - DNS
2013-01-16
boss (bind10)
socketcreator
msgq
cfgmgr
config store
cmdctl
stats daemon
ddns
xfrin
zonemgr
xfrout
auth
auth
auth
HTTP stats server
SNMP stats server
Data Source
Data Source
Data Source
loadzone
bindctl
Web UI
GUI
bind9 config
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

auth

- Authoritative DNS server

  DNS library
+ data sources
+ I/O
+ bit of logic

- Scales via multiple processes
  - Idea stolen from NSD


BIND 10 Process Model - DNS
2013-01-16
boss (bind10)
socketcreator
msgq
cfgmgr
config store
cmdctl
stats daemon
HTTP stats server
SNMP stats server
ddns
xfrin
zonemgr
xfrout
auth
auth
auth
Data Source
Data Source
Data Source
loadzone
bindctl
Web UI
GUI
bind9 config
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

xfrout

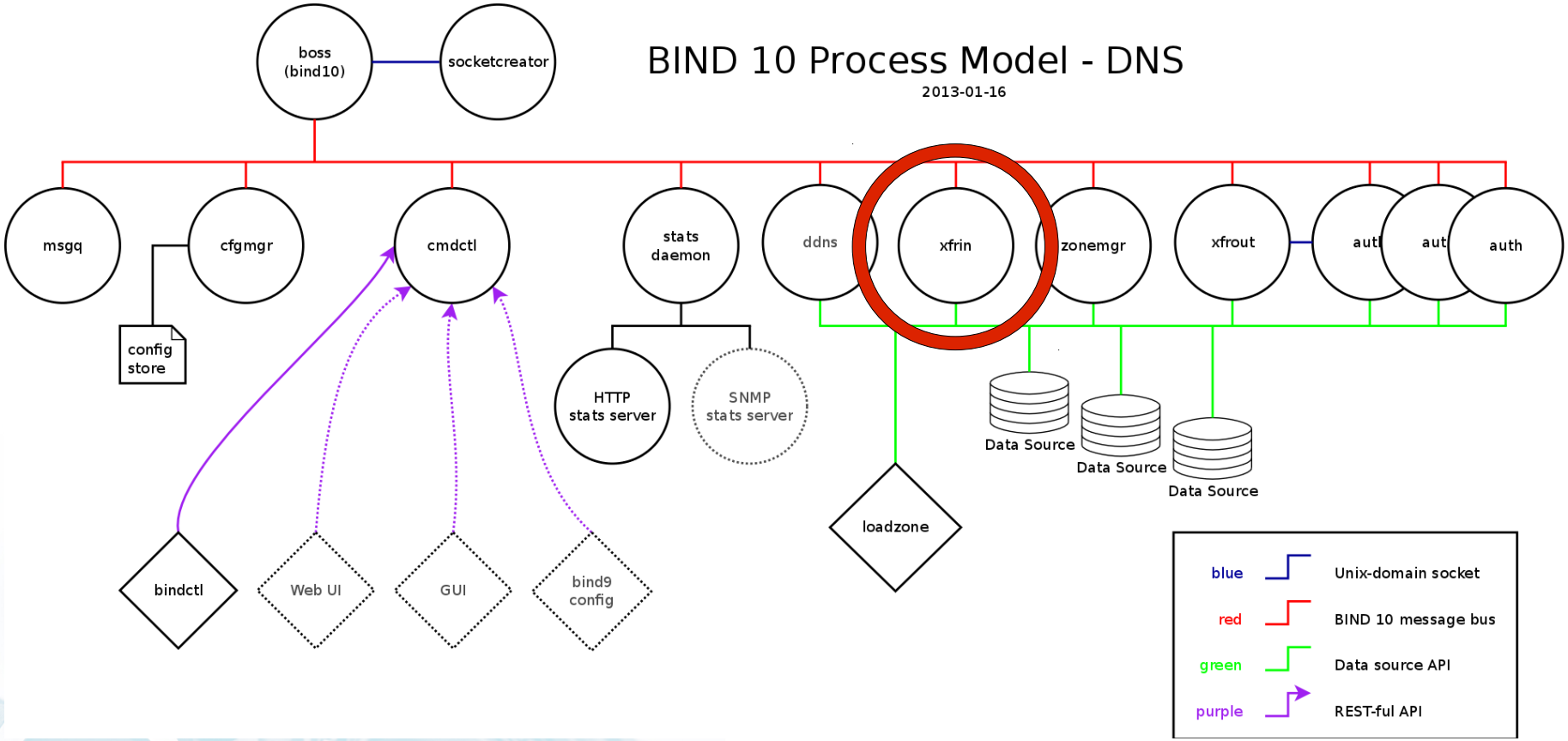
- AXFR/IXFR out, to act as a master

  DNS library
+ data sources
+ I/O
+ bit of logic

- Scales via multiple threads
  - New design with processes pending

A Final Aside: Passing Around Open Files

- AXFR/IXFR messages come to auth
- UDP packets can be forwarded
- TCP connections must go to xfrout
- Send file descriptor via `sendmsg()`:
  - `SOL_SOCKET`, `SCM_RIGHTS`
- Works on Linux, Solaris, BSD

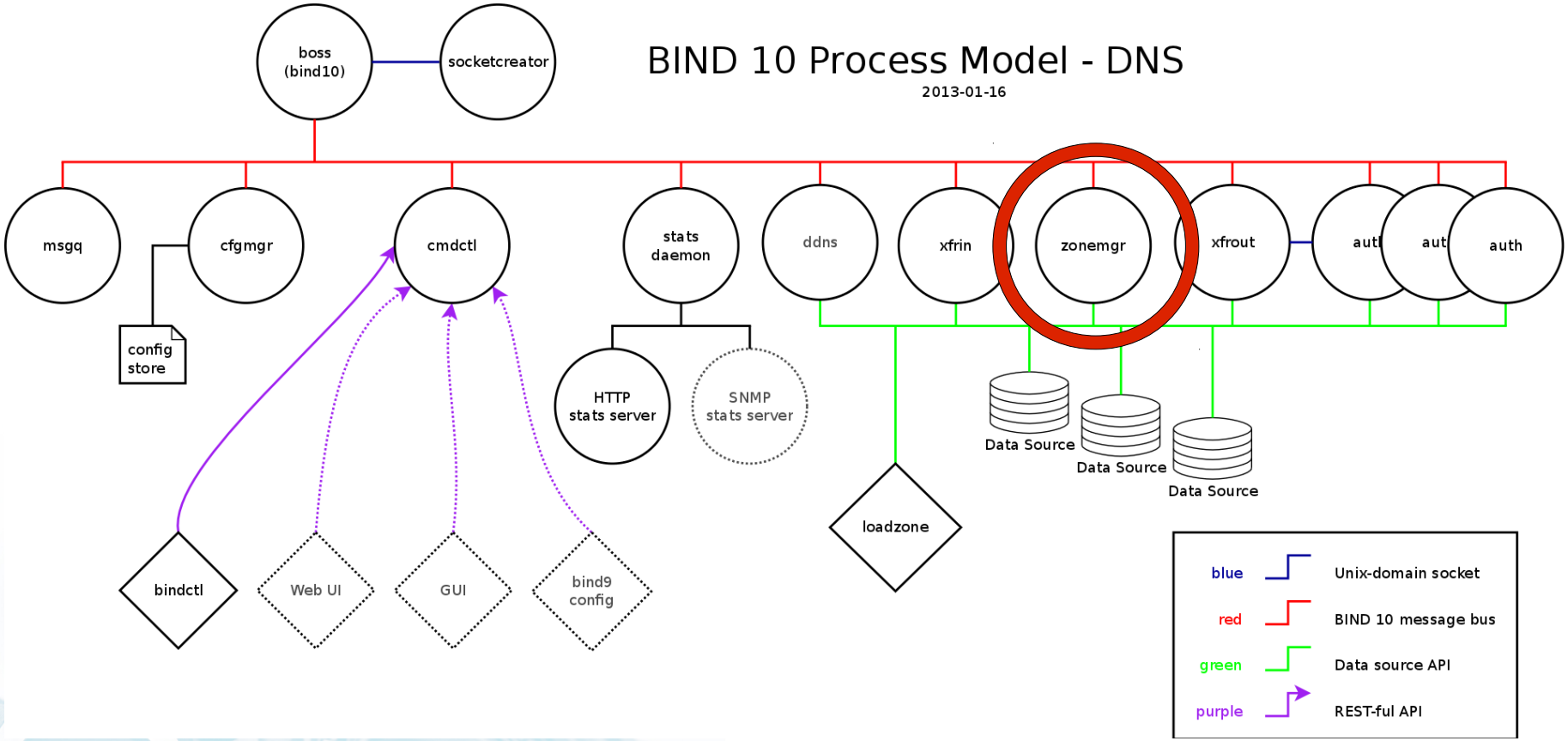

BIND 10 Process Model - DNS
2013-01-16
boss
(bind10)
socketcreator
msgq
cfgmgr
config
store
cmdctl
stats
daemon
HTTP
stats server
SNMP
stats server
ddns
xfrin
zonemgr
xfrout
auth
auth
auth
Data Source
Data Source
Data Source
loadzone
bindctl
Web UI
GUI
bind9
config
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

xfrin

- AXFR/IXFR in, to act as a slave

  DNS library (Python)
+ data sources
+ I/O
+ bit of logic

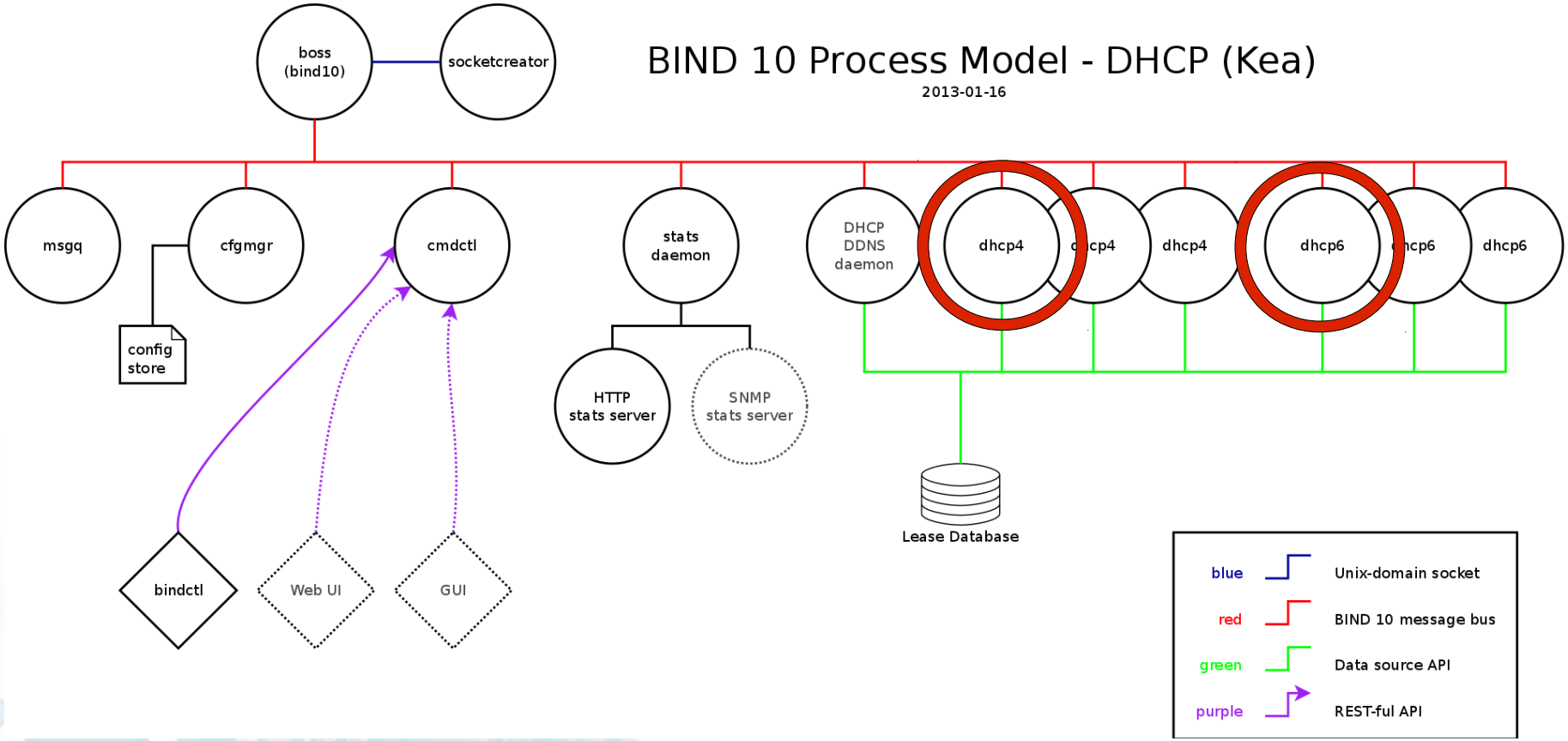
- Scales via multiple threads
  - New design with processes pending


BIND 10 Process Model - DNS
2013-01-16
boss (bind10)
socketcreator
msgq
cfgmgr
config store
cmdctl
stats daemon
HTTP stats server
SNMP stats server
ddns
xfrin
zonemgr
xfrout
auth
auth
auth
Data Source
Data Source
Data Source
loadzone
bindctl
Web UI
GUI
bind9 config
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

zonemgr

- Zone manager, times slave refresh
- Data sources
- That's it!
  - Will be collapsed into xfrin

Part the Third:
BIND 10 DHCP (a.k.a. *Kea*)


BIND 10 Process Model - DHCP (Kea)
2013-01-16
boss
(bind10)
socketcreator
msgq
cfgmgr
config
store
cmdctl
stats
daemon
HTTP
stats server
SNMP
stats server
DHCP
DDNS
daemon
dhcp4
dhcp4
dhcp6
dhcp6
Lease Database
bindctl
Web UI
GUI
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

DHCP4/6 Daemon Processes

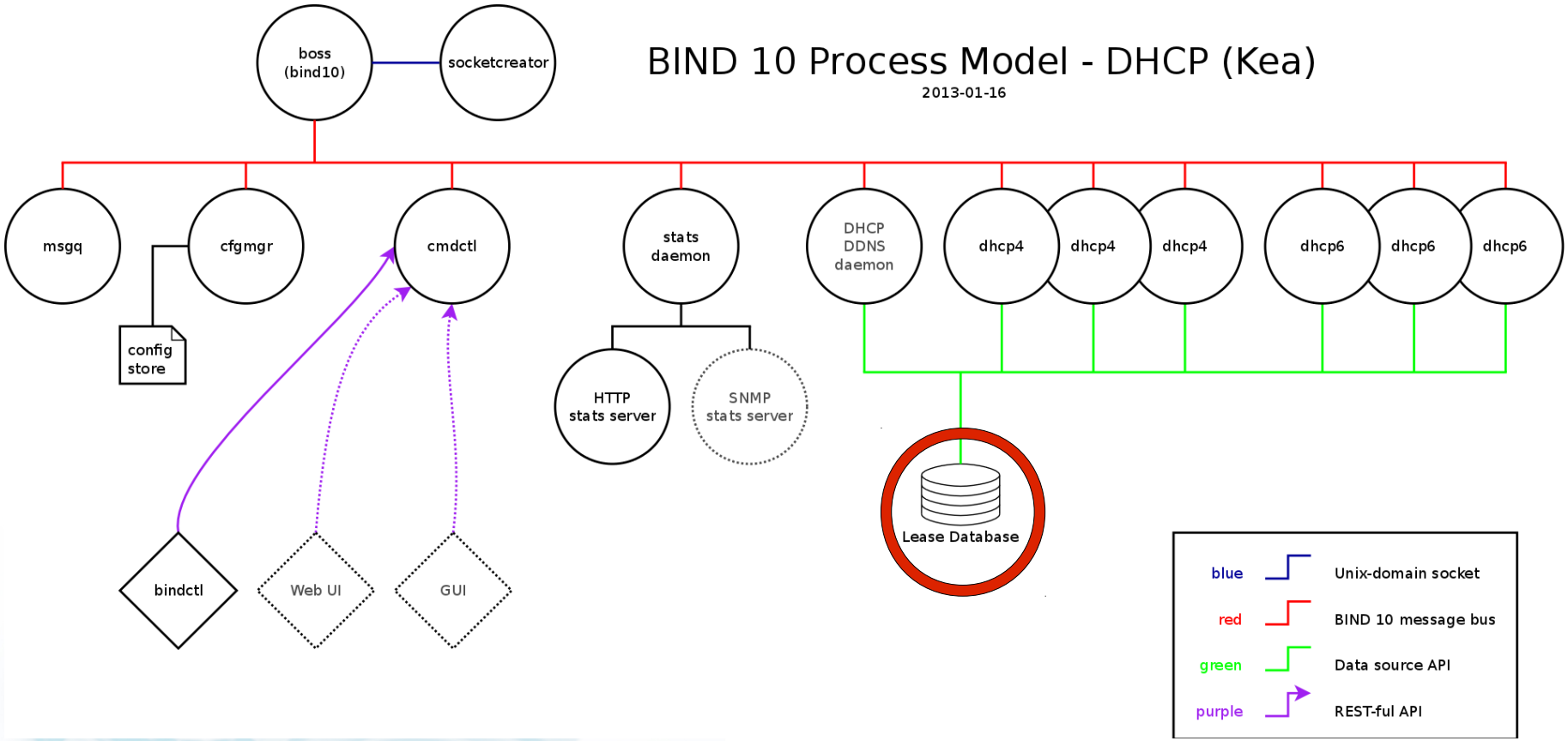
- Manage dynamic IPv4 and IPv6 address spaces
- Assign, renew, release IPv4 and IPv6 leases
- Assigns additional configuration options requested by IPv4/IPv6 hosts
- Dynamically reconfigurable

DHCP4/6 Multiple Processes

- Current thinking for scalability:
  - Divide queries between multiple processes
  - Receptionist process to route packets from a given client to the same daemon process to cope with state issues.
- Planned for 2013

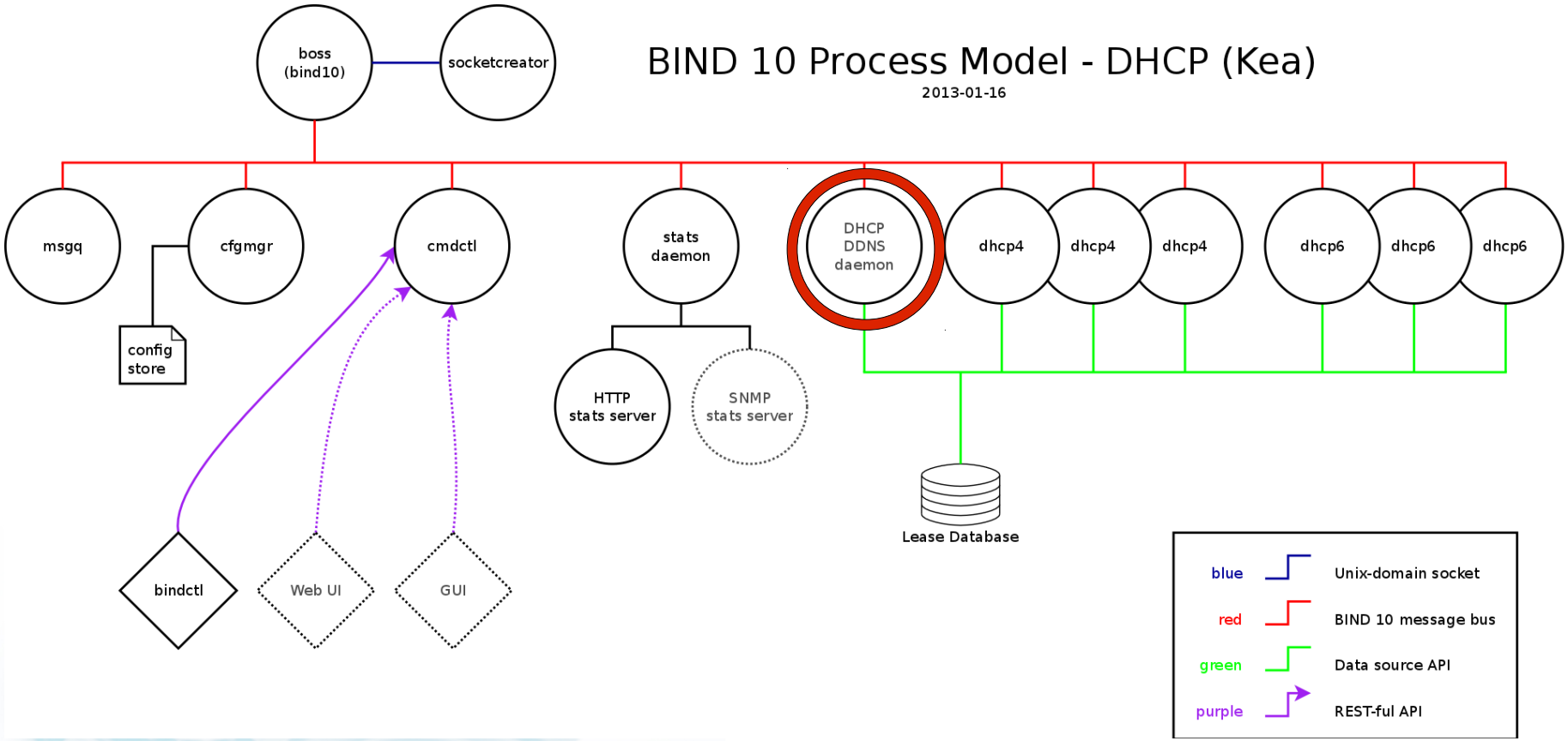
Supporting Library: libdhcp++

- Low-level DHCP messages (packets)
- C++ implementation


BIND 10 Process Model - DHCP (Kea)
2013-01-16
boss
(bind10)
socketcreator
msgq
cfgmgr
config
store
cmdctl
stats
daemon
HTTP
stats server
SNMP
stats server
DHCP
DDNS
daemon
dhcp4
dhcp4
dhcp4
dhcp6
dhcp6
dhcp6
Lease Database
bindctl
Web UI
GUI
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

Lease Database

- Leases held in a database
- Abstraction layer allows for different backends
  - MySQL currently implemented
  - In-memory backend available


BIND 10 Process Model - DHCP (Kea)
2013-01-16
boss
(bind10)
socketcreator
msgq
cfgmgr
config
store
cmdctl
stats
daemon
HTTP
stats server
SNMP
stats server
DHCP
DDNS
daemon
dhcp4
dhcp4
dhcp4
dhcp6
dhcp6
dhcp6
Lease Database
bindctl
Web UI
GUI
blue
Unix-domain socket
red
BIND 10 message bus
green
Data source API
purple
REST-ful API

DHCP DDNS Daemon

- Will handle addition/removal of name/address translations from forward and reverse DNS zones
- Implementation scheduled for 2013

DHCP Hooks

- Set of hooks to be included in the code:
  - Call out to user code at defined points in packet processing
  - Replaces “conditional” configuration processing in DHCP4
- API designed
- Implementation scheduled for 2013

http://bind10.isc.org/wiki/DhcpHooks

perfdhcp

- Utility to measure performance of DHCP servers
- Simulates multiple clients
- Measures round-trip time and throughput.

http://bind10.isc.org/wiki/DhcpBenchmarking