Cligen

Command-Line Interface Generator

Playing around with syntax

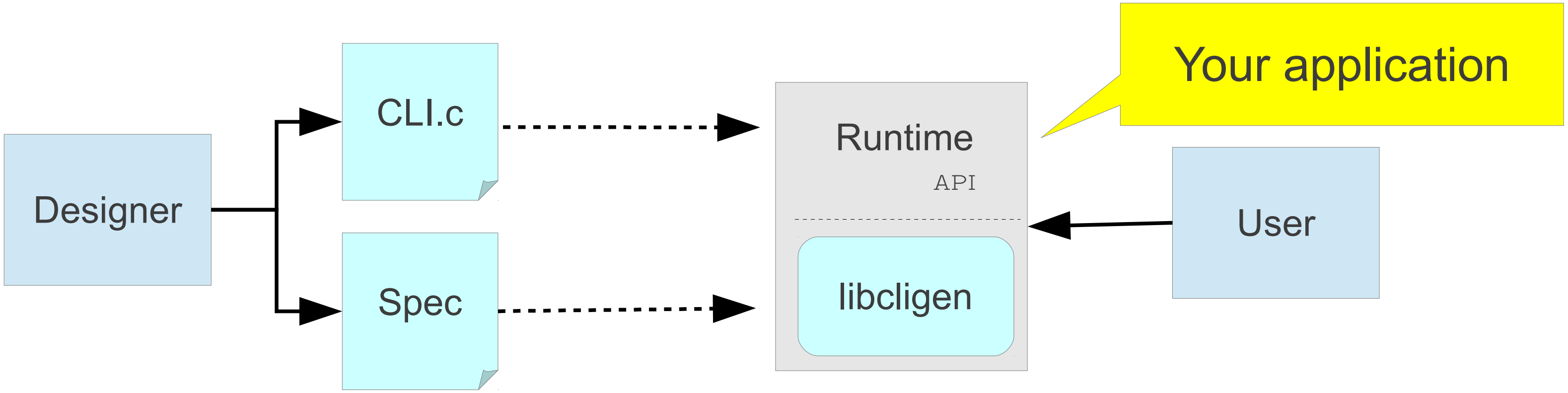
Olof Hagsand

```
T = A [b|c] @T, cb(42);
```

Background

- Designed systems and played around many projects
  - Dynarc – routers
  - Prosilient – measurement systems
  - RNR – router control plane
  - etc.
- Often spent lots of work with designing the interactive command-interface: CLI, XML, etc.
- Since 2001 I have therefore worked with Cligen for automatically generating textual interfaces

Idea



- Actually, cligen is not really a generator
- It is more a run-time system.
- You need to specify the CLI syntax and callbacks in C
- Then you link a library with your application and load the syntax.

Spec

```
prompt="hello> ";                 # Assignment of prompt
hello("Greet the world"){         # ’hello’ command with help text
    world, cb("Hello World!"); # ’world’ command with callback
}
```

```
> ./cligen_hello
hello> hello world
hello World!
```

CLI.c

```
int
hello_cb(cligen_handle h, varrec vars, cg_var *arg)
{
    printf("%s\n", arg->var_string);
    return 0;
}
```

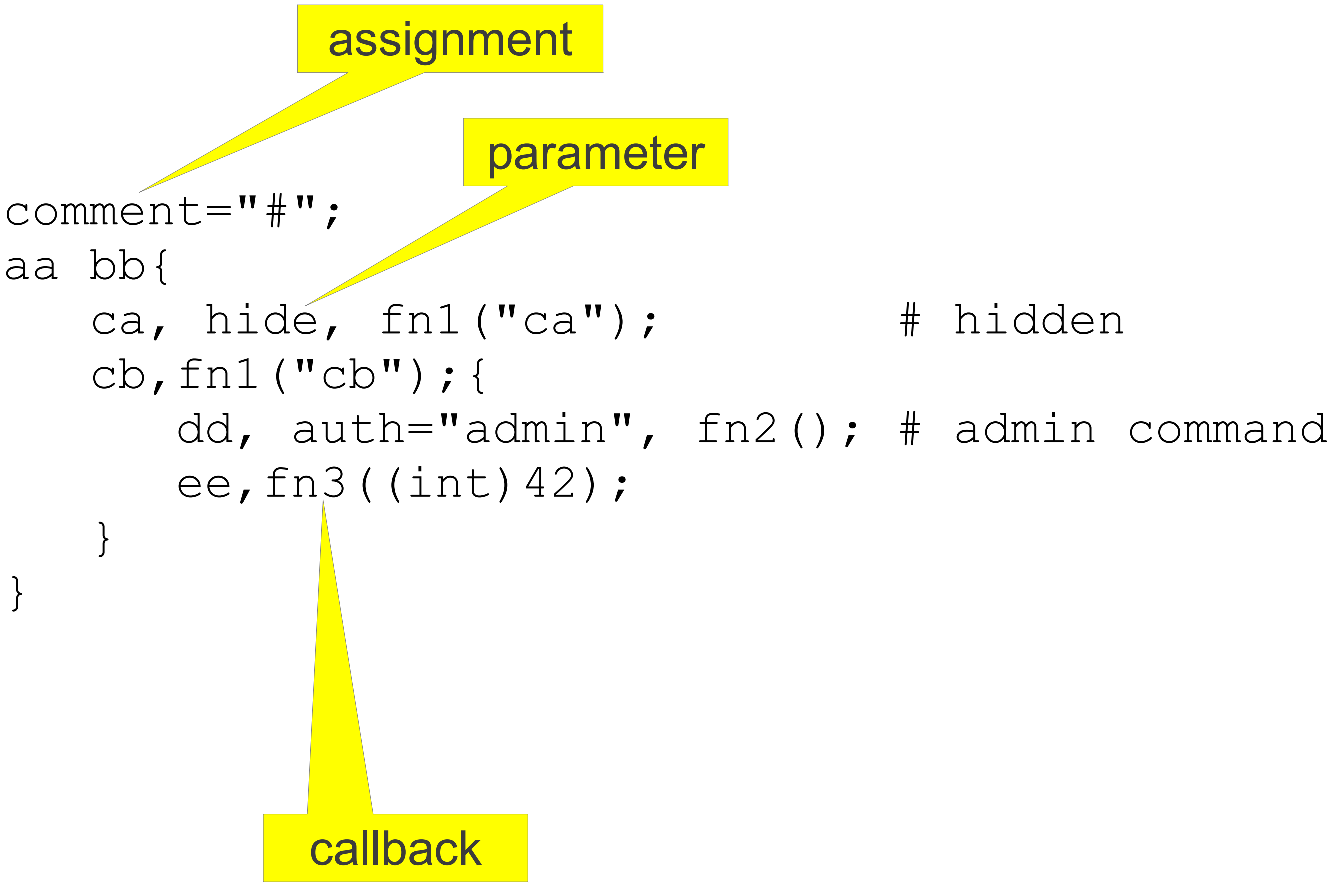
comment

variable

choice

```
ip("The IP protocol"){
   udp("The UDP protocol") <ipaddr>("IPv4 address");
   tcp("The TCP protocol") <int>("Port number");
}
```

```
cli> ip '?'
tcp        The TCP protocol
udp        The UDP protocol
cli> ip
```



```
comment="#";
aa bb{
   ca, hide, fn1("ca");        # hidden
   cb,fn1("cb");{
      dd, auth="admin", fn2(); # admin command
      ee,fn3((int)42);
   }
}
```

Static choice variable

```
interface <ifname type:string choice:eth0|eth1>("Interface name");
```

Dynamic choice variable (callback)

```
interface <ifname type:string interfaces()>("Interface name");
```

```
aa bb (ca("help ca")|cb("help cb")) [dd|ee];

aa bb{
      ca("help ca");{
         dd;
         ee;
      }
      cb("help cb");{
         dd;
         ee;
      }
   }
```

choice

Option choice

Alternative form

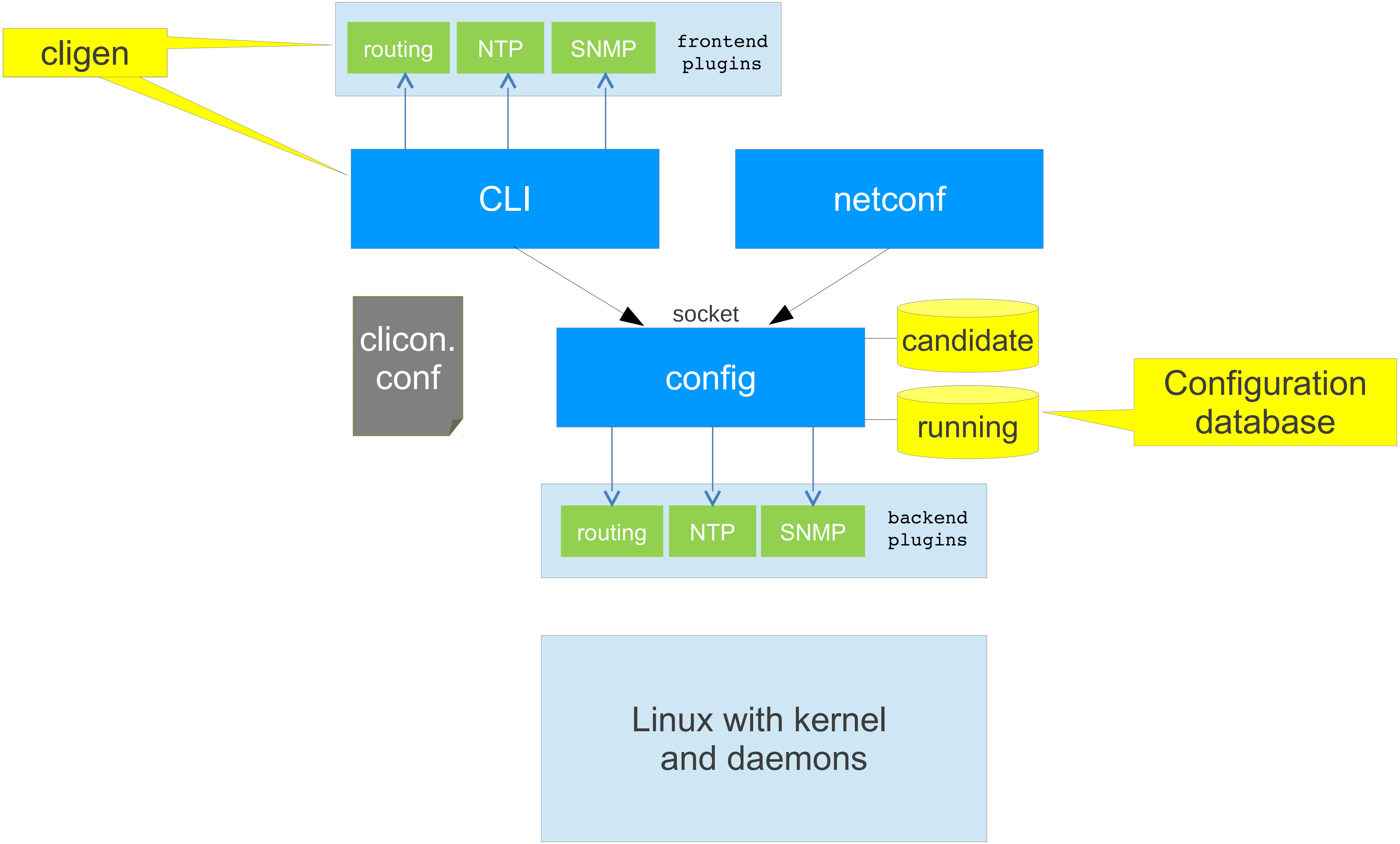
```
name="T";
interface <name type:string>{
    type (fddi | atm);
    description <freetext type:string>;
    media (auto | [full-duplex|half-duplex|simplex] [10M|100M]);
}

name="S";
set @T;
delete @T;
```

T is expanded within S

```
    name="T"; A [b|c] @T, cb();
```

Which are the valid commands?

Building a system - RNR



http://www.hagsand.se/cligen

3.1 is out.

3.2 on its way

cligen comes as GPL2 or commercial license