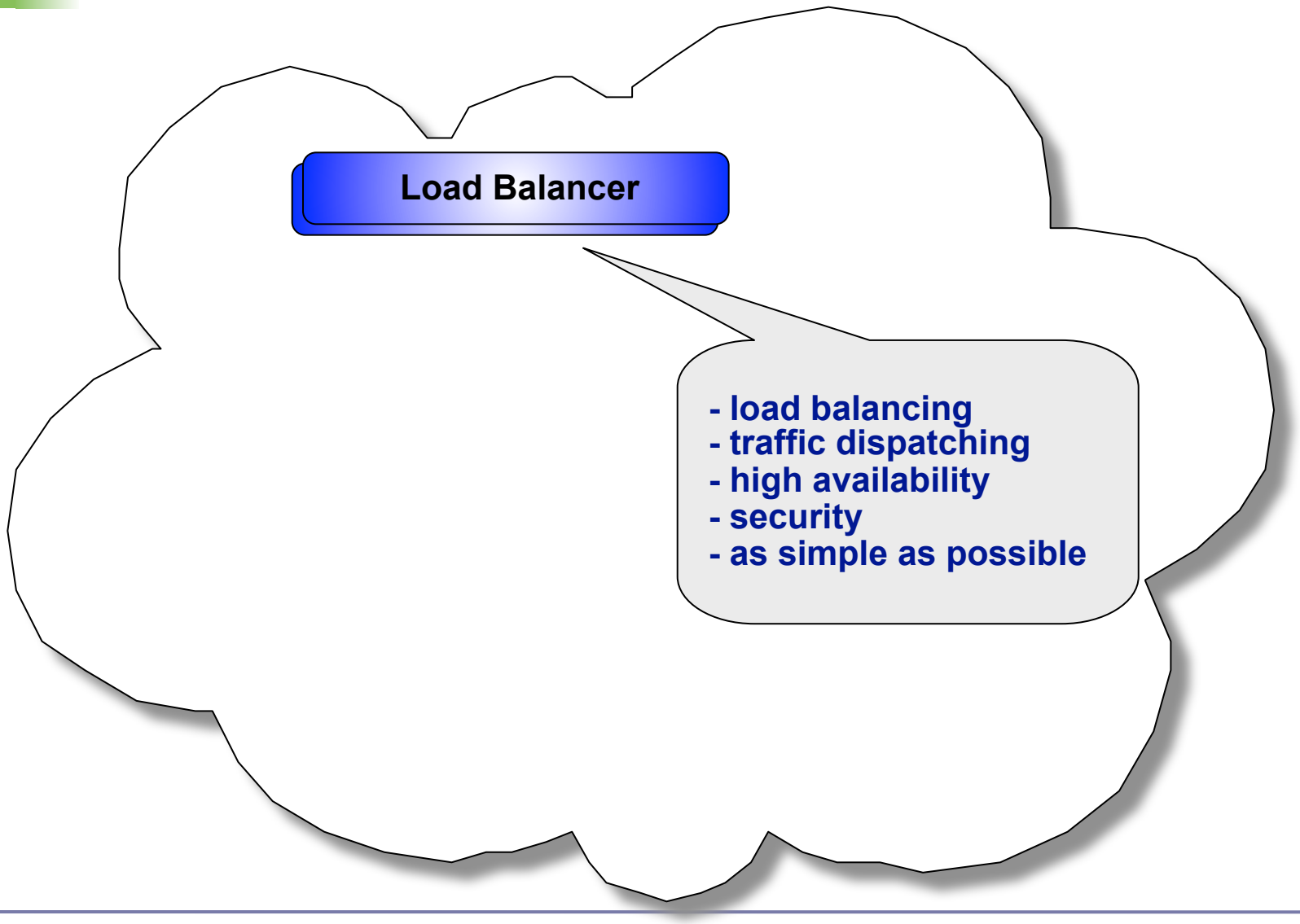
Load Balancing Load Balancers

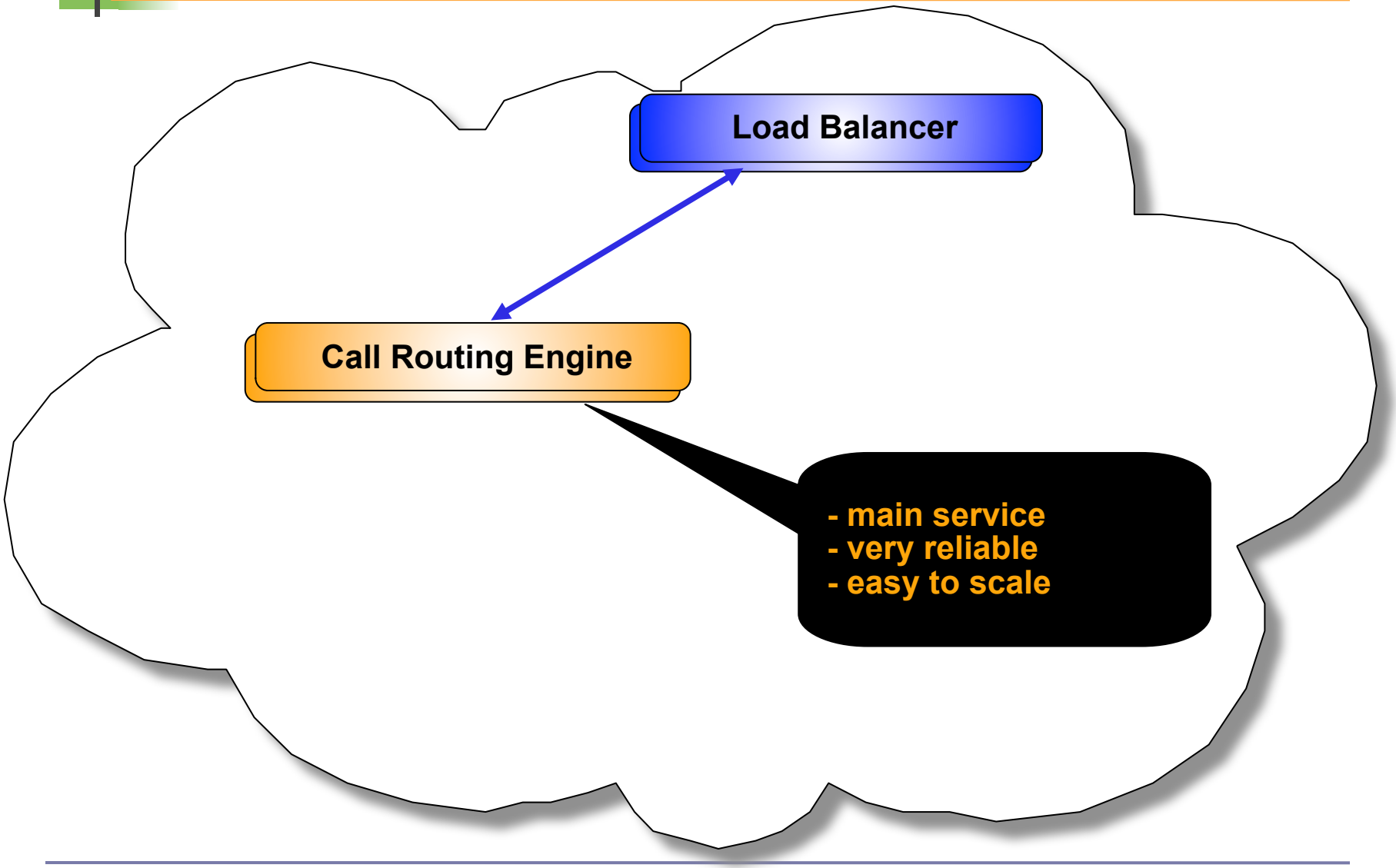
Daniel-Constantin Mierla
Co-Founder Kamailio Project

www.asipto.com

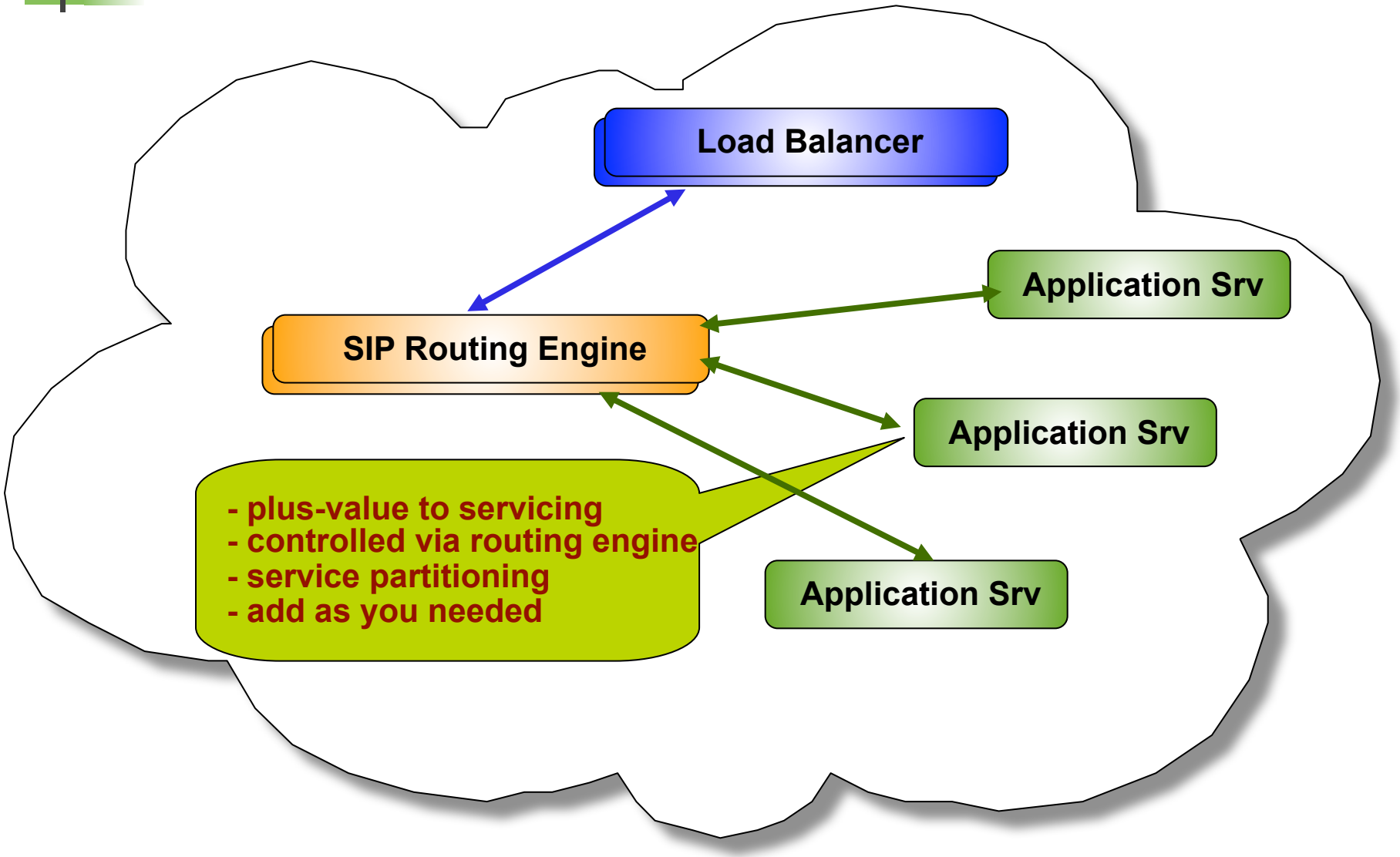
Load Balancing



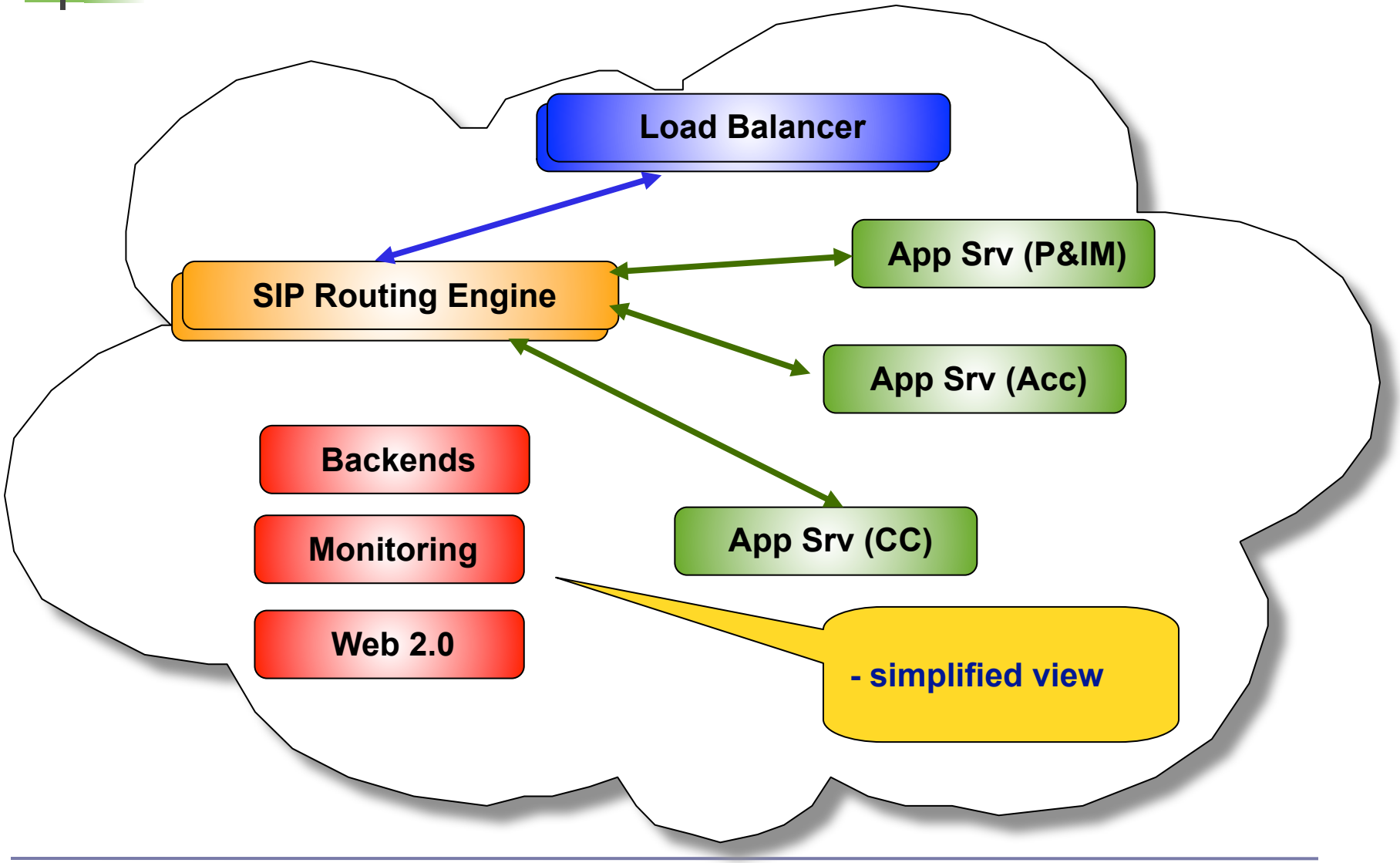
Load Balancing



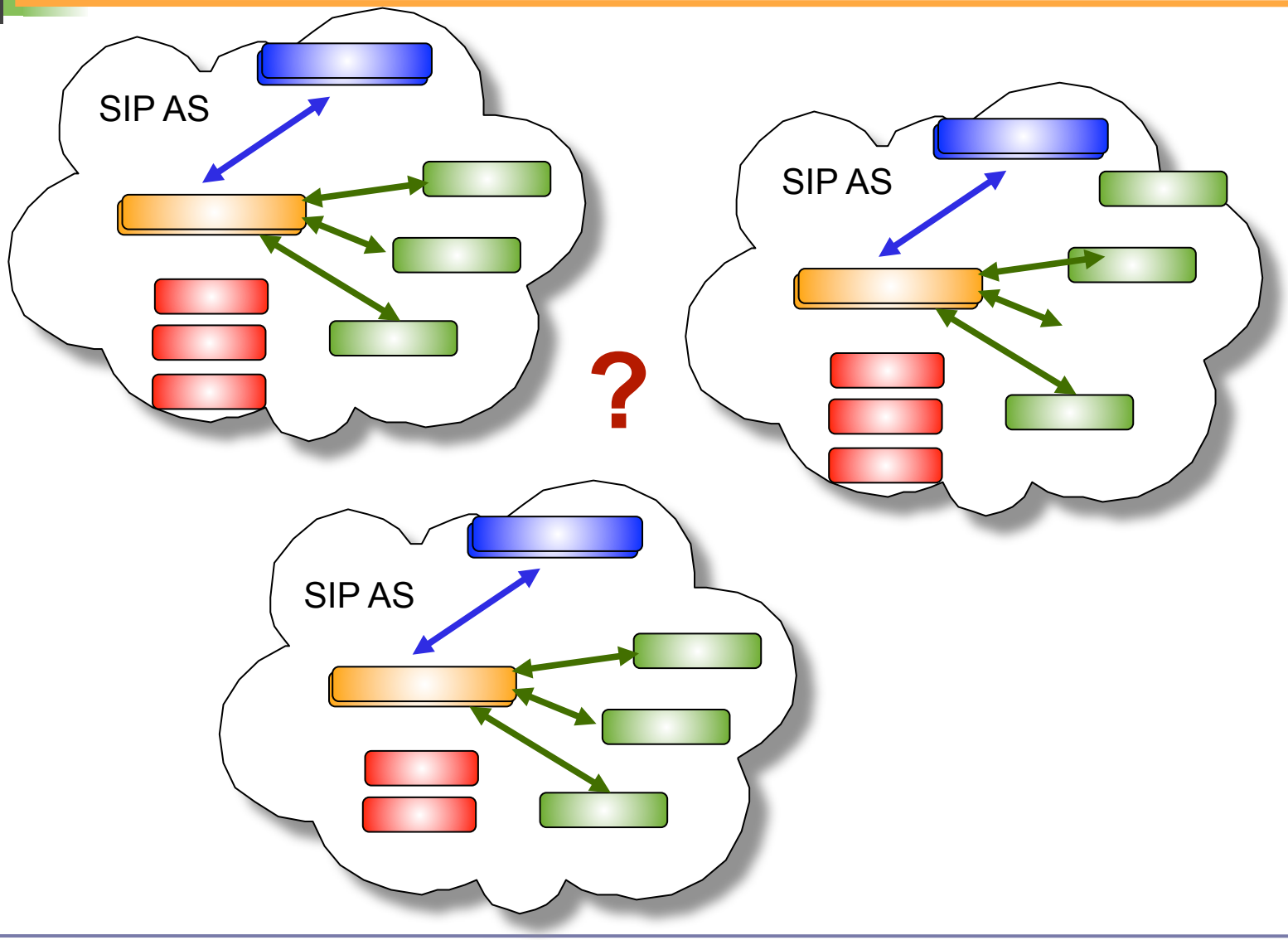
Load Balancing



Load Balancing



Load Balancing



Dispatcher Module - Load Balancing

- dispatching algorithms
  - compute an integer value using a hashing function

- hash over callid
  - ensures that all requests within a dialog goes to same box
- hash over from uri
  - ensures that all requests from same user goes to same box
- hash over to uri
  - ensures that registrations of an AoR goes to same box
- hash over request-uri
  - ensures that requests to same destination are processes by same box
- hash over config variable
  - for different needs

Dispatcher Module - Load Balancing

- balancing algorithms
  - call load distribution
    - counting of active calls per destination address
  - weight based distribution
    - each destination address has assigned a weight (percentage) to it
  - priority based distribution
    - each destination address has assigned a priority to it (practically is same as serial forking)
  - round-robin (next destination) - *recommended*
    - well-known algorithm with pretty fair distribution

Dispatcher destinations file

```
# setit(int) destination(sip uri) flags(int,opt) priority(int,opt) attributes(str,opt)

# proxies
2       sip:127.0.0.1:5080                          0   0   class=abc;strip=1;prefix=123
2       sip:127.0.0.1:5082;transport=tcp            0   0   class=xyz;strip=2;prefix=456

# gateways
1       sip:127.0.0.1:7070                          0   1
1       sip:127.0.0.1:7072;transport=sctp;px=3
1       sip:127.0.0.1:7074                          0   2
```

Dispatcher Config

```
# ----- dispatcher params -----
modparam("dispatcher", "db_url",
        "mysql://openser:openserro@localhost/openser")
modparam("dispatcher", "table_name", "dispatcher")
modparam("dispatcher", "flags", 2)
modparam("dispatcher", "dst_avp", "$avp(AVP_DST)")
modparam("dispatcher", "grp_avp", "$avp(AVP_GRP)")
modparam("dispatcher", "cnt_avp", "$avp(AVP_CNT)")
```

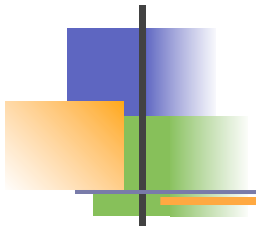
Dispatcher Config

```
# Dispatch requests
route[DISPATCH] {
     # round robin dispatching on gateways group '1'
     if(!ds_select_dst("1", "4"))
     {
          send_reply("404", "No destination");
          exit;
     }
     xlog("L_DBG", "--- SCRIPT: going to <$ru> via <$du>\n");
     t_on_failure("RTF_DISPATCH");
     t_relay();
     exit;
}
# Failure re-route
failure_route[RTF_DISPATCH] {
     if (t_is_canceled()) {
          exit;
     }
     # next DST - only for 500 or local timeout
     if (t_check_status("500")
               or (t_branch_timeout() and !t_branch_replied()))
     {
          if(ds_next_dst())
          {
               t_on_failure("RTF_DISPATCH");
               t_relay();
               exit;
          }
     }
}
```

Dispatcher Module - Features

- simple, lightweight extension - fast and reliable
- can combine load balancing and dispatching algorithms as needed in the same configuration file
- can work with flat text file or database for loading routing records
- can reload routing records at runtime (no need to restart)
- can auto-detect out-of-service destinations
- can ping destination to automatically inactivate/activate
- offers high flexibility in configuration file
  - *event routes, access to attributes and list of destinations*
- can detect if traffic is coming from an address in the destinations list

Scaling the load balancer



Nobody got to the upper limits of Kamailio routing performances!!!

Thank You, That's All!

... anyhow ...

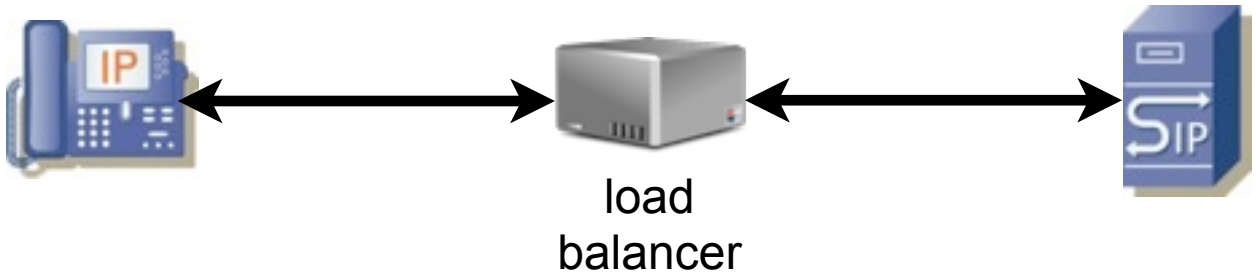
... let’s do it just for research purposes ...

Dispatcher Config - Stateful forwarding

```
# Dispatch requests
route[DISPATCH] {
     # round robin dispatching on gateways group '1'
     if(!ds_select_dst("1", "4"))
     {
          send_reply("404", "No destination");
          exit;
     }
     xlog("L_DBG", "--- SCRIPT: going to <$ru> via <$du>\n");
     t_on_failure("RTF_DISPATCH");
     t_relay();
     exit;
}
# Failure re-route
failure_route[RTF_DISPATCH] {
     if (t_is_canceled()) {
          exit;
     }
     # next DST - only for 500 or local timeout
     if (t_check_status("500")
               or (t_branch_timeout() and !t_branch_replied()))
     {
          if(ds_next_dst())
          {
               t_on_failure("RTF_DISPATCH");
               t_relay();
               exit;
          }
     }
}
```

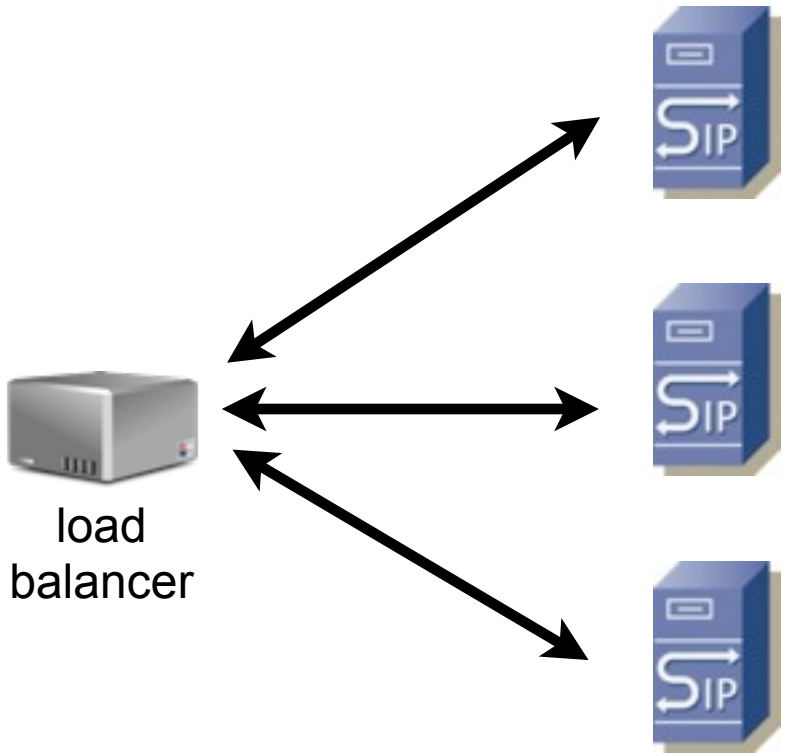
Dispatcher Config - Stateless forwarding

```
# Dispatch requests
route[DISPATCH] {
     # round robin dispatching on gateways group '1'
     if(!ds_select_dst("1", "4"))
     {
          send_reply("404", "No destination");
          exit;
     }
     xlog("L_DBG", "--- SCRIPT: going to <$ru> via <$du>\n");
     forward();
     exit;
}
```



DNS SRV Load Balancing

```
_sip._udp.kamailio.com. IN SRV   10 1 5060 sip1.kamailio.com.
_sip._udp.kamailio.com. IN SRV   10 1 5065 sip2.kamailio.com.
_sip._udp.kamailio.com. IN SRV   10 1 6065 sip3.kamailio.com.
```



Load Balancing to /dev/null

```
request_route {
        ;
}
```

Load Balancing to One Host

```
request_route {
    forward(1.2.3.4);
}
```

```
request_route {
    rewritehostport("1.2.3.4");
    forward();
}
```

Stateless Config Round Robin Load Balancer

Per Process Round Robin Routing!

```
loadmodule “sl.so”
loadmodule “textops.so”
loadmodule “pv.so”
modparam("pv", "varset", "i=i:0")
request_route {
    if(!is_method(“INVITE”)) {
        sl_send_reply(“404”, “Not Found”);
        exit;
    }
    $var(i) = ($var(i) + 1 ) mod 2;
    if($var(i)==1) {
        rewritehostport(“1.2.3.4”);
    } else {
        rewritehostport(“2.3.4.5”);
    }
     forward();
}
```

Stateless Config Round Robin Redirect Balancer

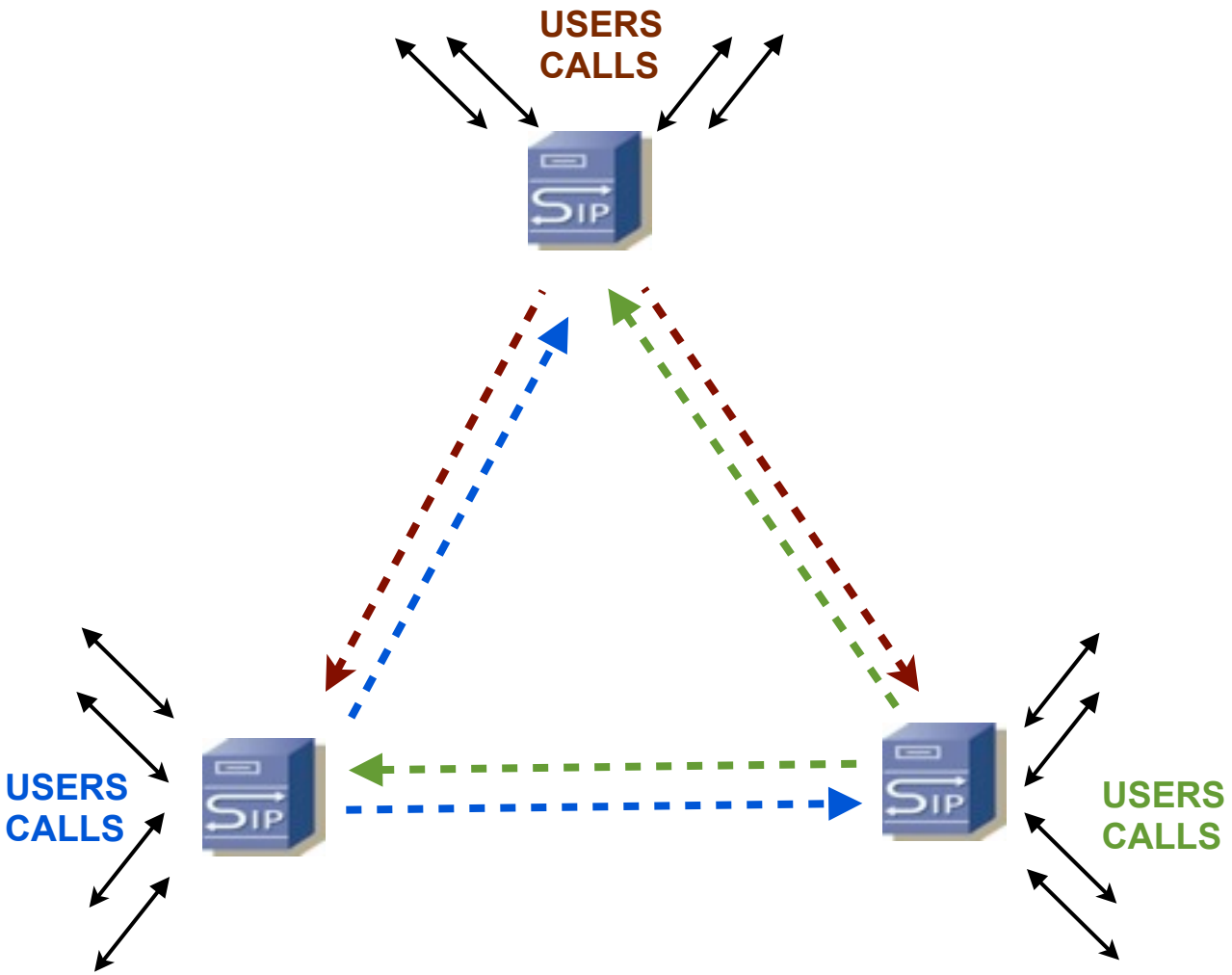
```
loadmodule “sl.so”
loadmodule “textops.so”
loadmodule “pv.so”
modparam("pv", "varset", "i=i:0")
request_route {
    if(!is_method(“INVITE”)) {
        sl_send_reply(“404”, “Not Found”);
        exit;
    }
    $var(i) = ($var(i) + 1 ) mod 2;
    if($var(i)==1) {
        rewritehostport(“1.2.3.4”);
    } else {
        rewritehostport(“2.3.4.5”);
    }
     sl_send_reply(“301”, “Moved Temporarily”);
}
```

Stateless Config Round Robin Load Balancer

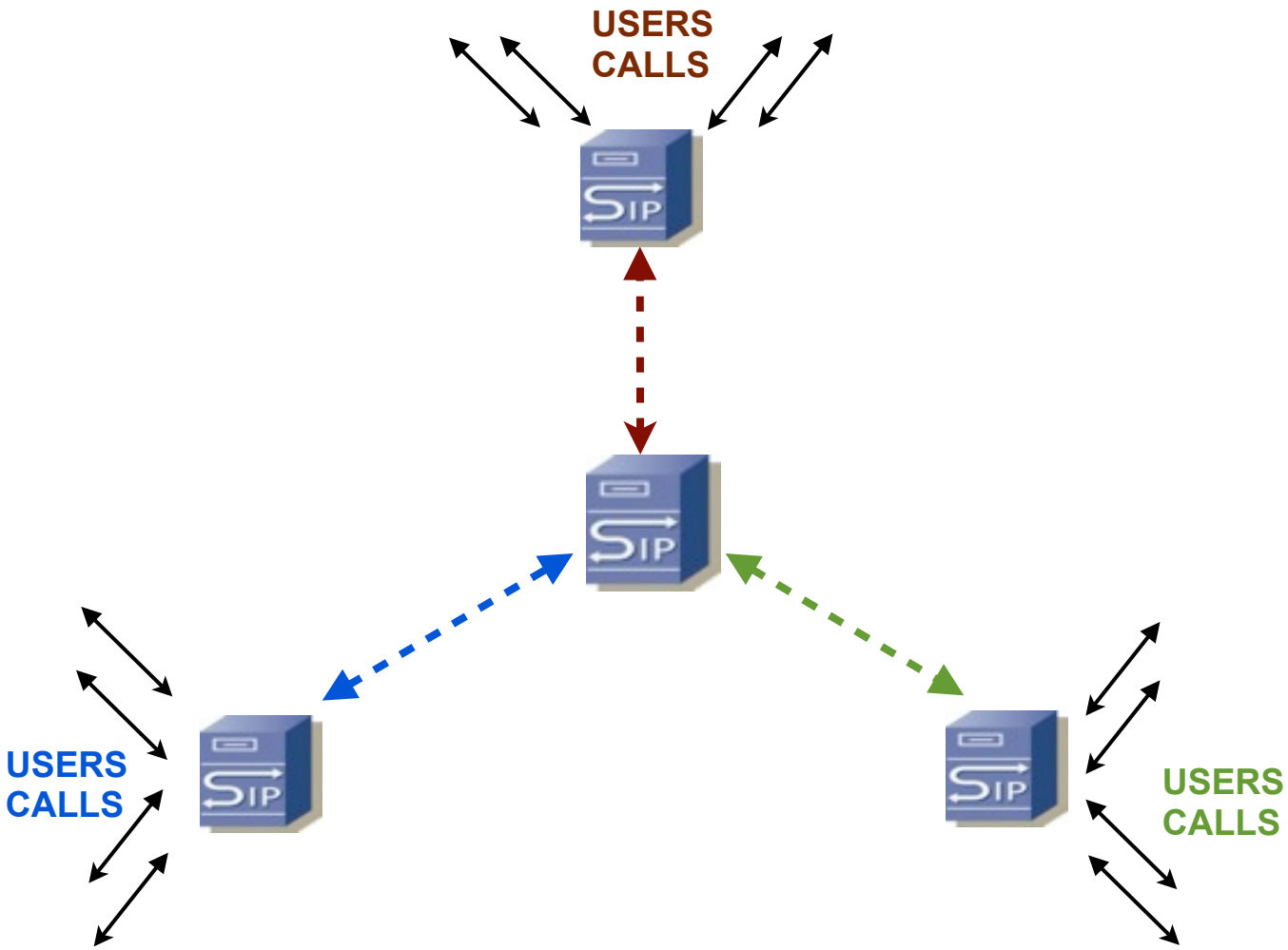
```
loadmodule “sl.so”
loadmodule “textops.so”
loadmodule “pv.so”
loadmodule “cfgutils.so”
modparam("pv", "shvset", "i=i:0")
modparam("cfgutils", "lock_set_size", 1)
request_route {
    if(!is_method(“INVITE”)) {
        sl_send_reply(“404”, “Not Found”);
        exit;
    }
    lock(“balancing”);
    $shv(i) = ($shv(i) + 1 ) mod 2;
    $var(x) = $shv(i);
    unlock(“balancing”);
    if($var(x)==1) {
        rewritehostport(“1.2.3.4”);
    } else {
        rewritehostport(“2.3.4.5”);
    }
    forward();
}
```

Per Instance Round Robin Routing!

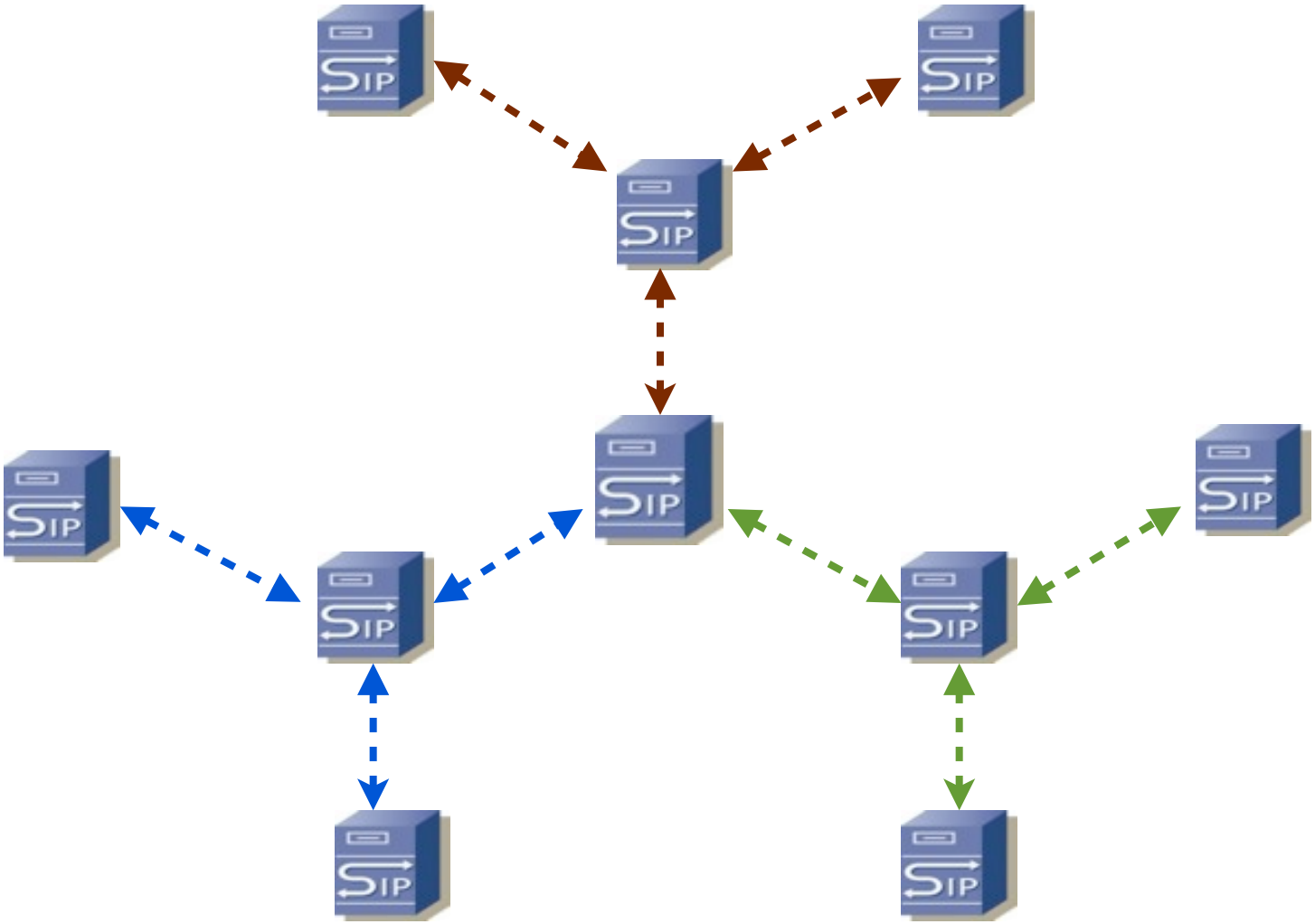
Scalability - Paritioning



Scalability



Scalability



??? questions ???

www.asipto.com
@miconda

sponsors and exhibitors