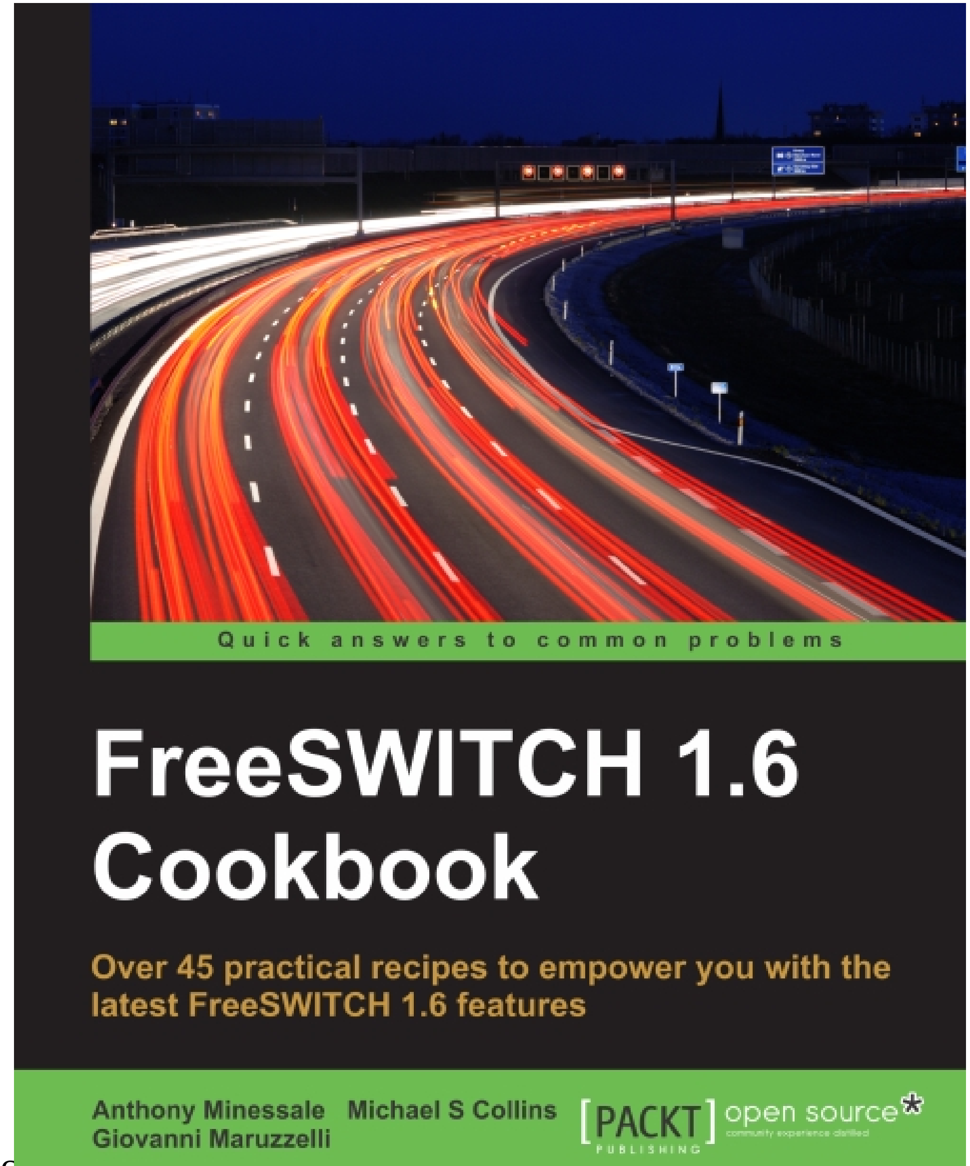
Load Balancing FreeSWITCHes

Giovanni Maruzzelli
gmaruzz@OpenTelecom.IT

OpenTelecom.IT
opensource telecommunication

www.packtpub.com

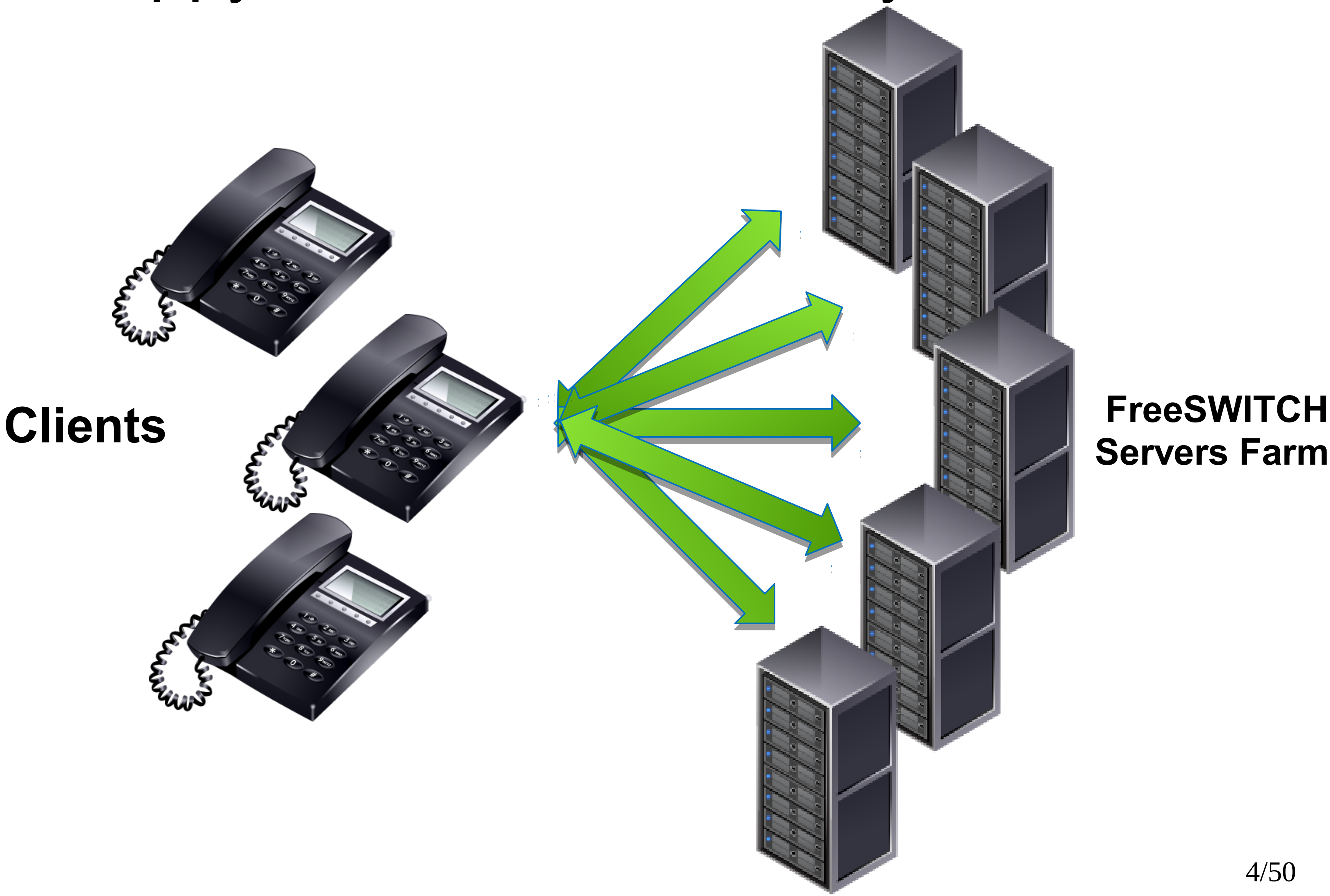
25% Discount Code: **CLUECON25**



Agenda

- Different options and strategies to load balancing FreeSWITCHes, using Kamailio, OpenSIPS or FreeSWITCH itself: each one has its own unique advantages, both for horizontal scaling and for HA resilience.
- We will go through definition of problems and analysis of solutions, and how to implement each platform using best practices.

Happy Problems, Too Many Clients...



Load Balancing Techniques

- DNS (clients go to different IPs)
  - Round Robin
  - SRV Records

- HUB (clients go to same IP)
  - (SER) Kamailio
  - (SER) OpenSIPS
  - FreeSWITCH

DNS (clients go to different IPs)

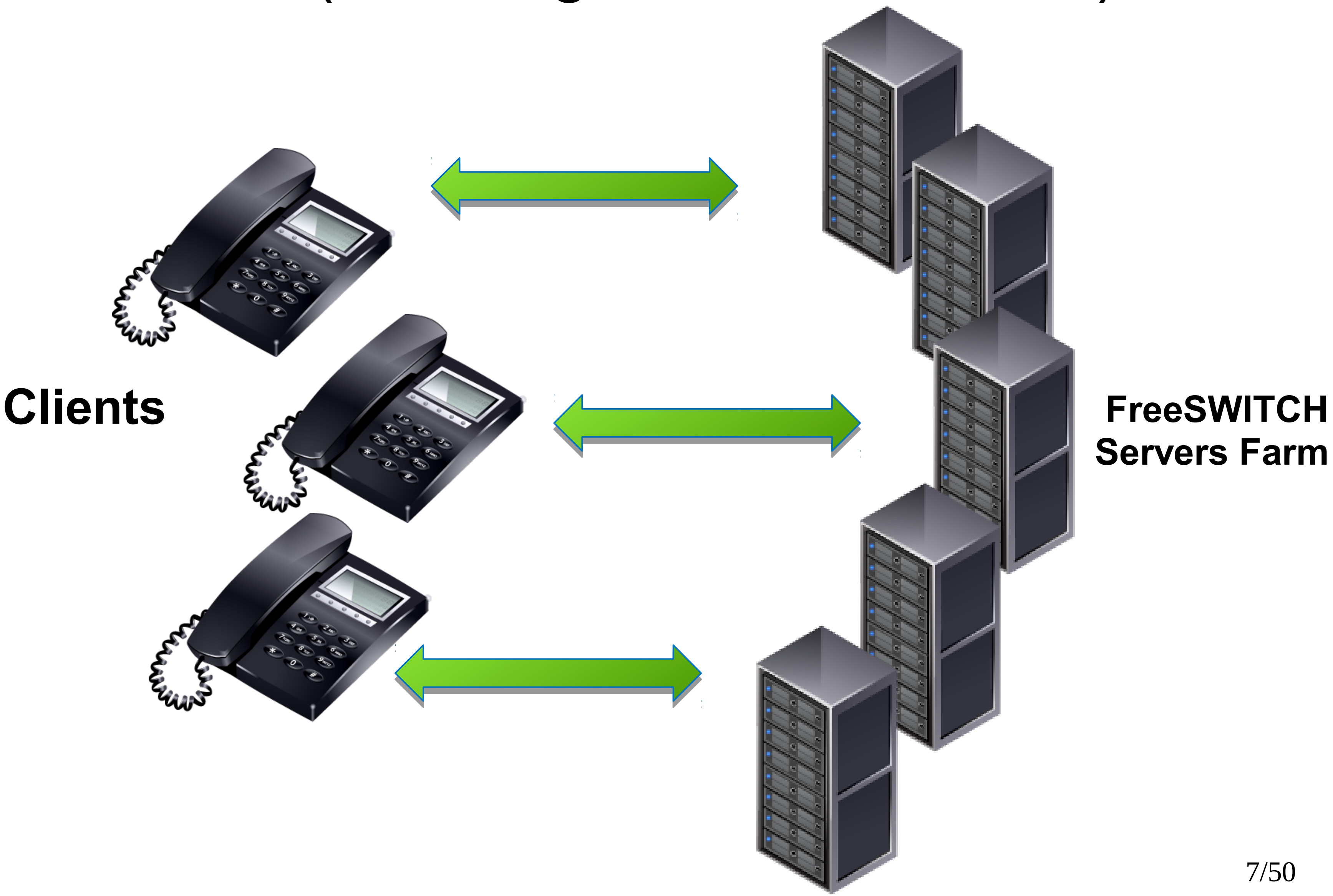
- Round Robin
  - More IPs for same name, alternates

- TTL
  - With a short TTL, you can disable failed servers

- SRV records
  - Priority
  - Weight
  - Client try next one on failed server

- DNS SIP probing, failover + low TTL
  - Route 53

DNS (clients go to different IPs)

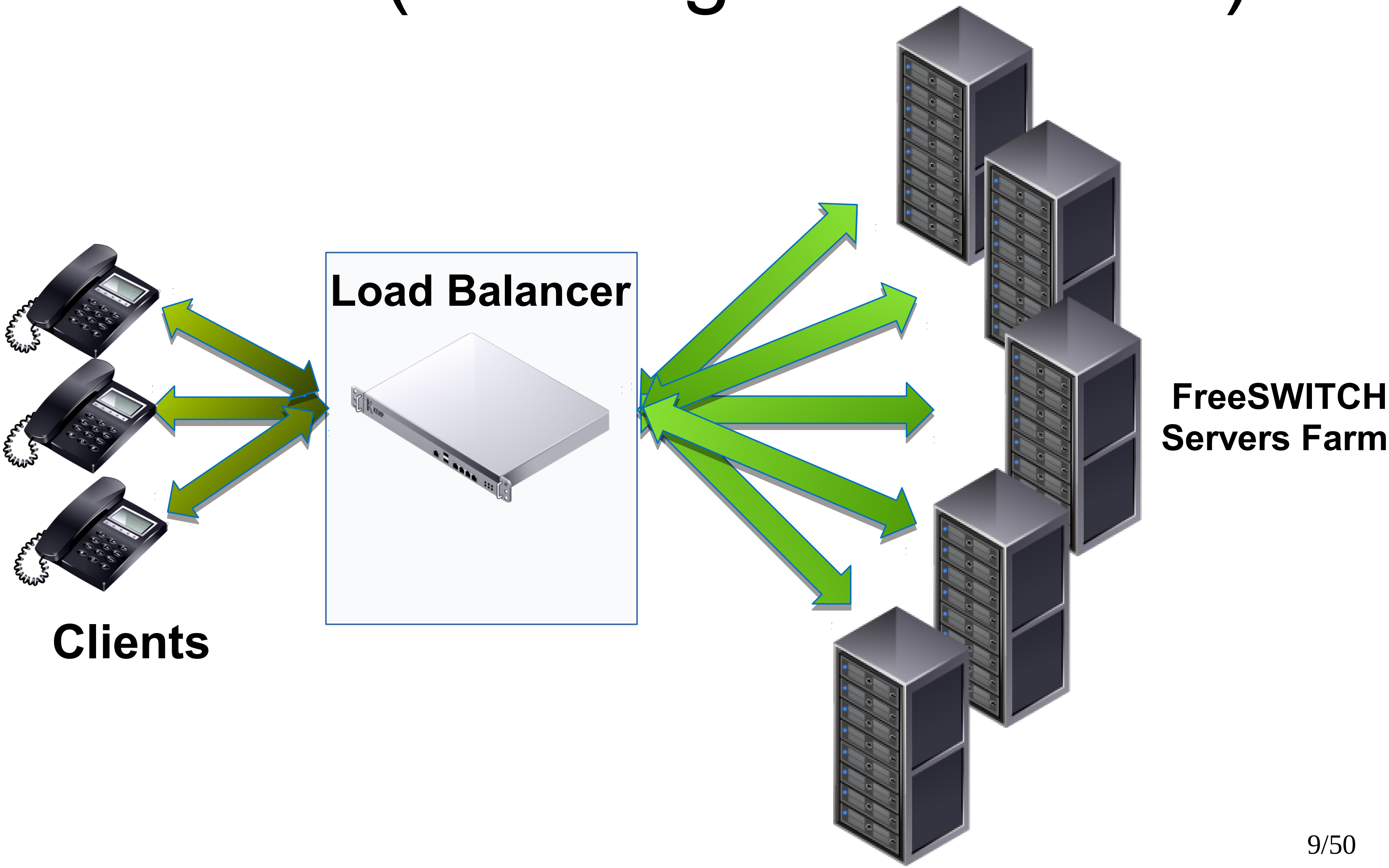


HUB (clients go to same IP)

- SER (kamailio, opensips)
  - SIP proxies (pass requests and responses)
  - Very Performing and Stable
  - Very Low Level (Deep SIP Knowledge Required)
  - Fire and Forget

- FreeSWITCH
  - SIP B2BUA (answer requests giving responses)
  - Enough Performer (media-bypass, media-proxy)
  - High Level (Do The Right Thing)
  - Ready Made Building Blocks

HUB (clients go to same IP)



What is your VoIP server doing?

- Signaling
- Media

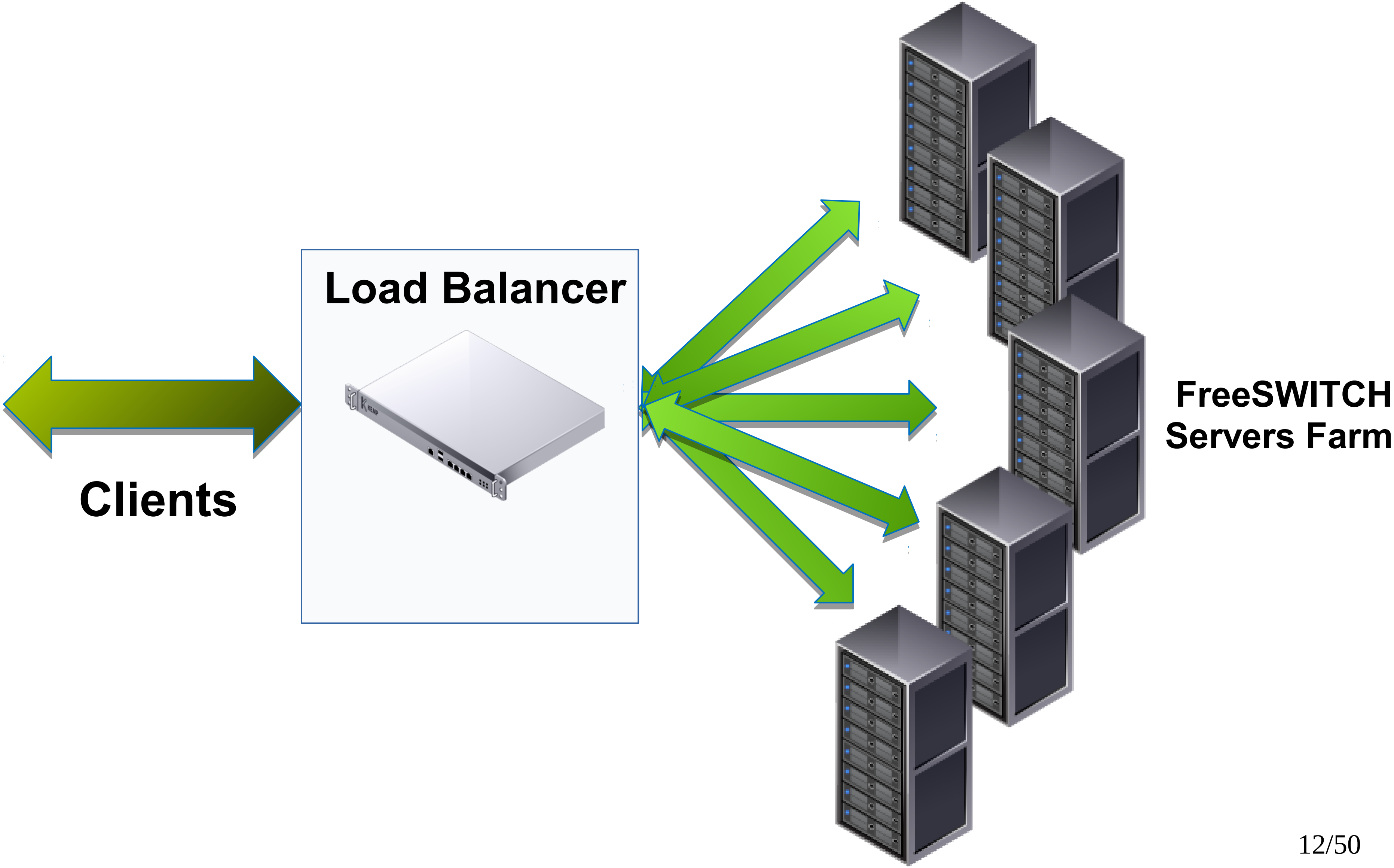
- WebRTC
- Web pages
- NAT traversing (eg: relaying audio)

- Registration (Location)
- NAT piercing (eg: OPTIONS)

IPv4 Location NEEDS HUB

- Client is behind NAT
- Client sends from its own IP:port a REGISTER request to Location Server IP:port, and in doing so it opens a pinhole in the NAT, waiting for server's answer
- NAT pinhole is only able to receive packets from same IP:port icouple (Client/Server) it was open by, and for a limited period of time (30 seconds?)
- Location Server sends periodically from same IP:port an OPTIONS message to Client IP:port, Client answers, and in doing so it maintains the pinhole open
- When there is an incoming call for Client, Server sends the INVITE from same IP:port to Client IP:port
- (**YES**, you can build logic to send INVITE from the server to which callee is registered to, but is not ready made)

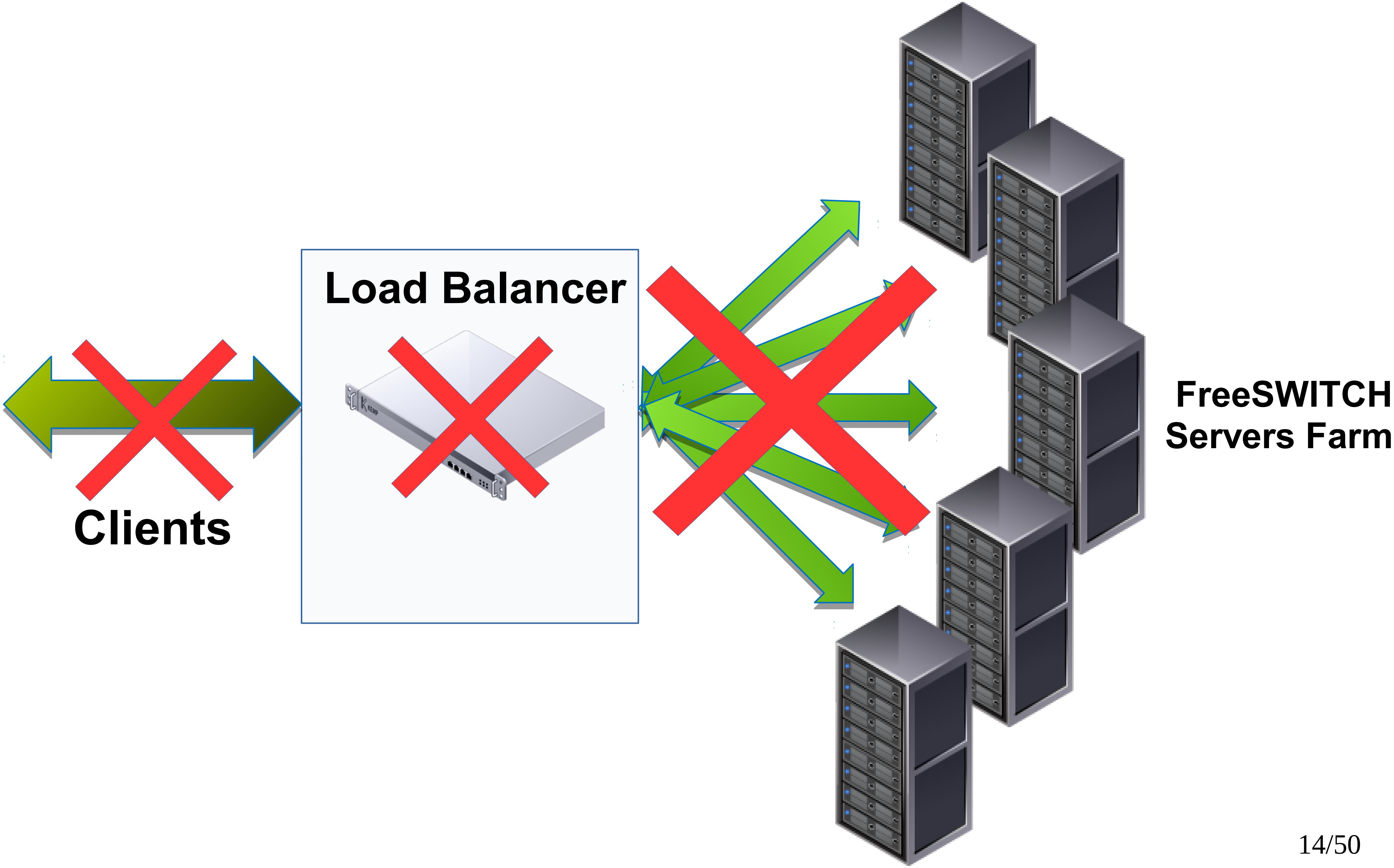
So, what is HUB Load Balancing about ?



Horizontal Scalability

- Distribute requests on two or more resources:
    - Registrations
    - Calls
    - Transcoding
    - Voice Mail
    - PSTN Termination and/or Origination
    - Audio Conferencing
    - Video Muxing
    - IVR
    - Web Pages
    - Web Sockets

What is HUB Load Balancing NOT about ?

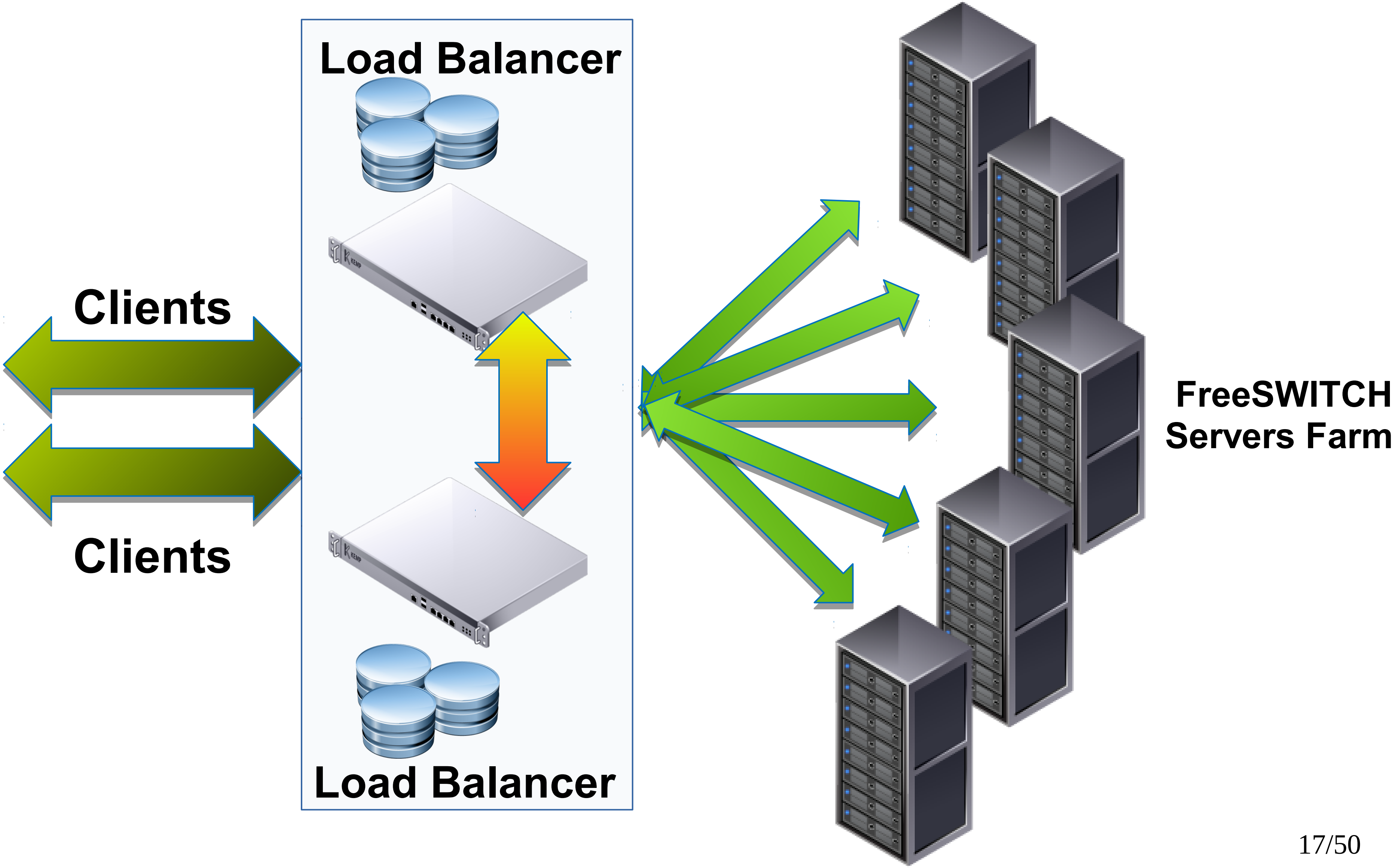


High Availability, NOOOOOT

- If anything happens to Load Balancer's
  - Cabling
  - Switch
  - Ethernet Card
  - Power Supply
  - Disk
  - RAM
  - DataBase
- LB as Single Point of Failure
- Only one entrance (and/or exit)


VERY NICE
GREAT SUCCESS

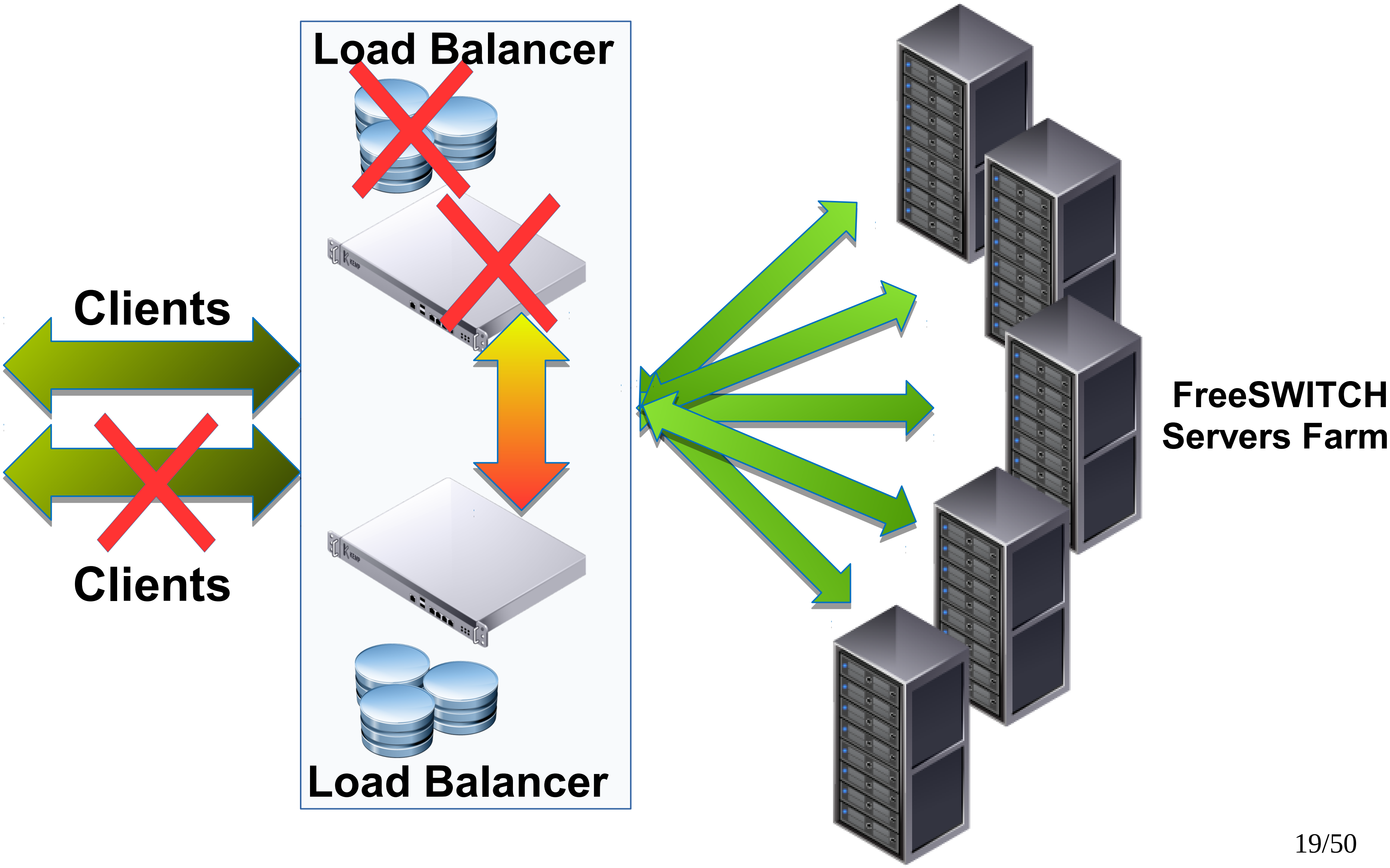
HIGH AVAILABILITY (HA) LB



HA techniques: double it all

- LAN Switch and Cabling
- Load Balancer
  - Virtual (Floating) IP address
  - HeartBeat, Keepalived, Corosync
- File System
  - DRBD
  - NFS
  - Rsync
- Database
  - Cluster
  - Master-Master (Active-Passive)

HA LB on Failure



Proxy and B2BUA, as Load Balancers

- **Proxy** (Kamailio and OpenSIPS)**:**
  - Pass along the signaling (Requests and Responses), back and forth
  - Stateful or Stateless
  - Can balance (distribute) Registrations

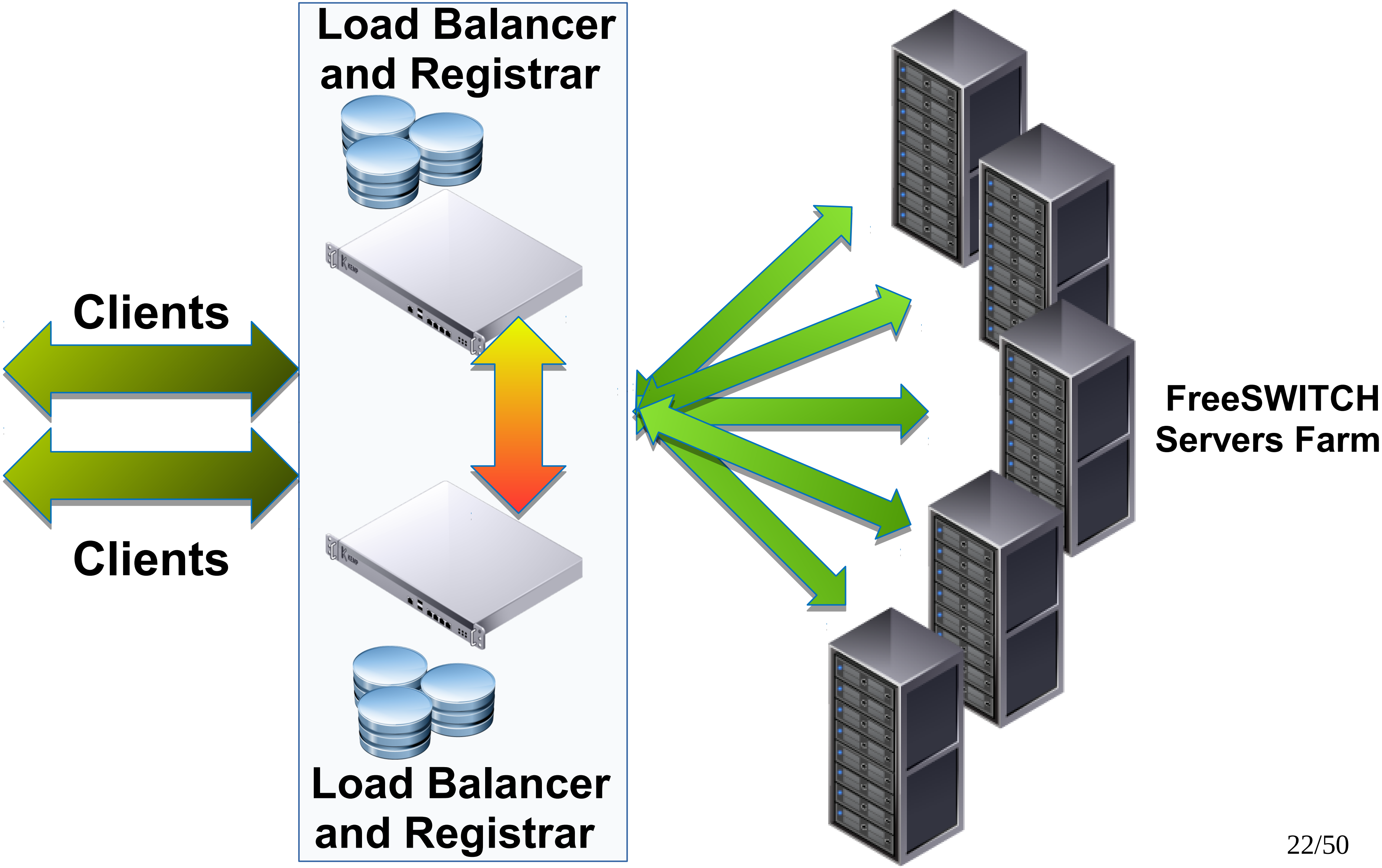
- **B2BUA** (FreeSWITCH)**:**
  - Gives Responses to Requests, and generates new Requests
  - Answers a Call, generates another Call, bridges those two into one audio flow
  - Cannot balance (distribute) Registrations

Where to put the REGISTRAR

- **ON LB MACHINE**, directly interacting with Clients
  - Both Kamailio, FreeSWITCH and OpenSIPS support Registrar on LB machine
  - REGISTER and NAT Keepalive (OPTIONS, NOTIFY) are high volume, low load transactions
  - One robust box (in active-passive HA) will be able to serve tens of thousands clients

- **ON SEPARATE MACHINES**, load balanced by LB
  - FreeSWITCH cannot load balance registrations
  - FreeSWITCH can act as registrar, load balanced by Kamailio or OpenSIPS
  - This topology scales indefinitely: partitioning, redirect

STANDARD (HA) LB and REGISTRAR



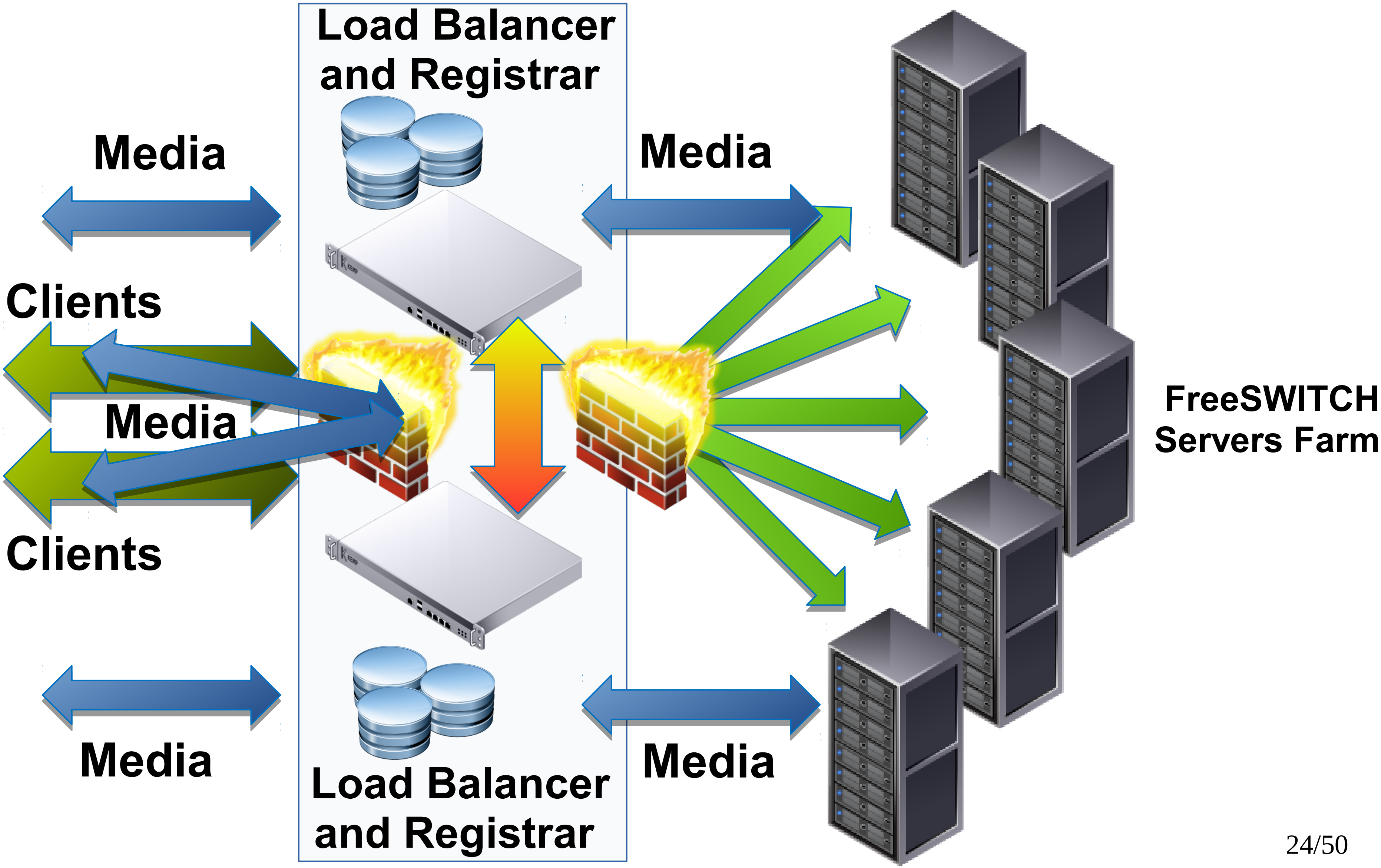
Load Balancer (and SBC?) Additional Features

(...and what is EXACTLY an SBC, anyway?)

- SECURITY
  - Limits (eg, per account)
  - DOS and DDOS (flood)
  - IP Addresses Ban
  - Topology Hiding

- NAT traversing - Media RELAYING
- Transport TRANSLATING (udp, tcp, tls, dtls, rtp, srtp)
- Media TRANSCODING (g711, opus, v8, h264)

AIM: (HA) LB, "SBC" and REGISTRAR



SER Limit: DIALOG

- Dialog module in Kamailio and OpenSIPS allows you take trace of active calls
- In the initial transaction (INVITE) you create the “dialog”. Dialog is destroyed by BYE, or timeout
- Dialogs can be grouped in “profiles”, based on SIP and derived variables: from, to, account, ...
- You can limit how many dialogs (calls) are allowed based on each of those values, and combination of them, eg: number of concurrent calls to Oceania from each “Purple” account

FS Limit: “limit” and “limit_execute”

- “limit” is in default mod_dptools and mod_commands
- You use it in dialplan and scripting to check numbers of concurrent “whatevers”: calls, resources, applications executions, variables, ...
- Its counters are incremented when “limit” is called, and decremented when current call ends (or by uuid_limit_release)
- It has many different backends: Hash, DB, Redis
- You can use it to limit number of concurrent calls by user, destination, account, gateway, etc, and for limiting resource accesses or checking external counters
- Surpassing “limit” transfers the call to the extension “limit_exceeded” (and decrements), while surpassing “limit_execute” continues without executing the application

SER Flood: PIKE, Ratelimit

- DOS, Fraud: Kamailio and OpenSIPS can use Pike module for limiting the number of calls per second for each source IP
- DDOS: ratelimit can be used to limit max number of transactions per second for each transaction type, eg: INVITE (calls), REGISTER, BYE, etc

FS Flood: Limit

- “limit”, when used with “Hash” backend, accepts a “period” argument, during which the specific counter applies. Eg: a limit of “120” each “60” seconds, or of “2” each “1” second. THEY'RE DIFFERENT LIMITS !
- “Hash” backend is local to the server, but API “hash_remote” allows for cumulative synchronization of a pool of FreeSWITCH servers

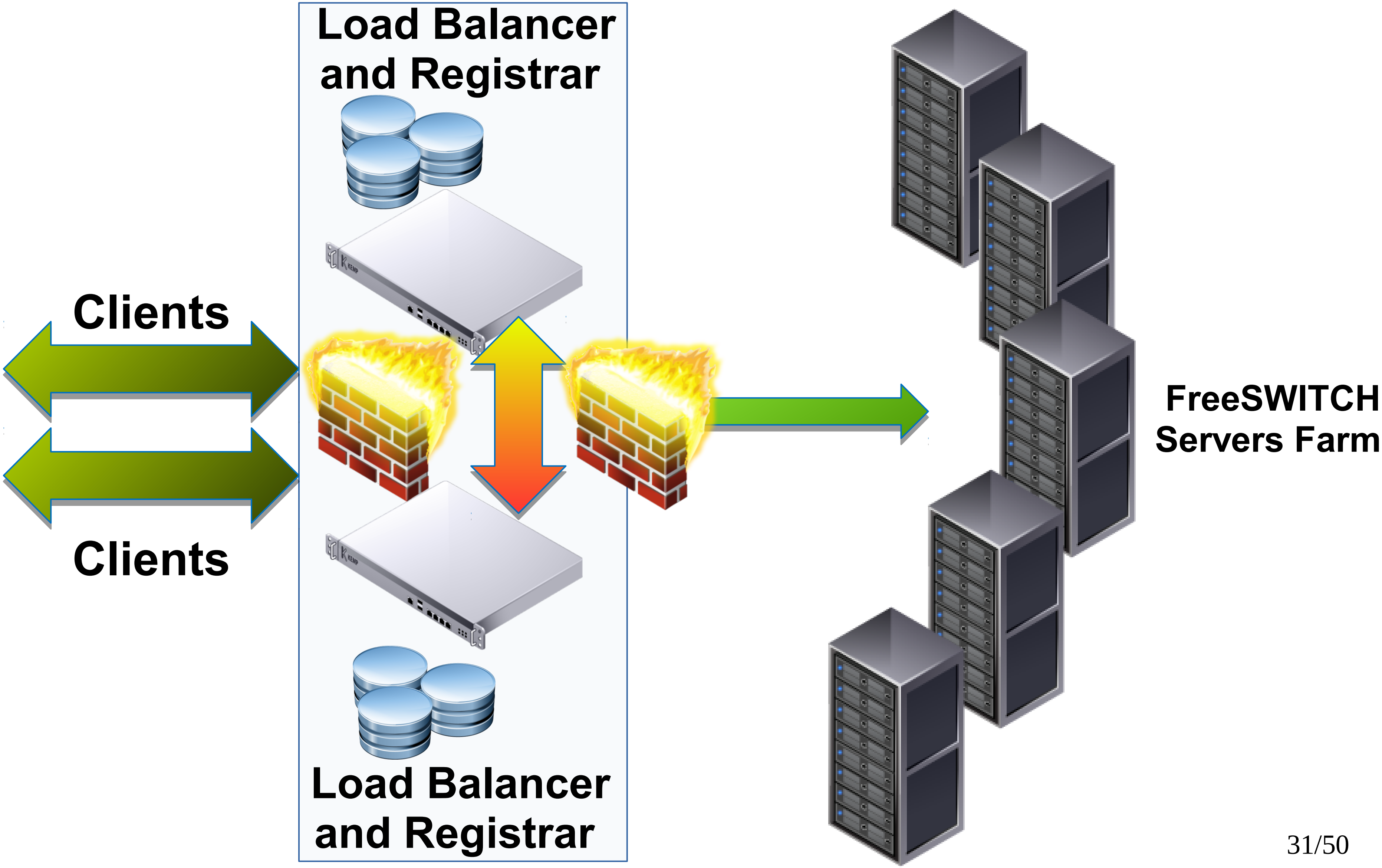
FS and SER: Ban IP Addresses

- To completely stop malicious, buggy, looped, or misconfigured clients to flood your service, only way is to block them for good at the IP/Firewall level
- You can use IPTables scripting on Linux (a famous one was written by Kris Kielhofner, google for it), manipulate firewall rules from FreeSWITCH and SER application's scripts, or use Fail2Ban
- Easiest and proven way, is to use Fail2Ban

Ban IP addresses with Fail2Ban

- Fail2Ban originally written to lock you out after multiple failed login retry
- Based on continuous monitoring of a logfile, checks the number of “fails” for time period
- Easily adaptable to “whatever” checking: you just needs to write a “fail” line in a file
- In FS, you can use “log” API, while in SER the “xlog” config script instruction
- For monitoring failed registration in FS, you can use the ready made mod_fail2ban

SECURITY Features



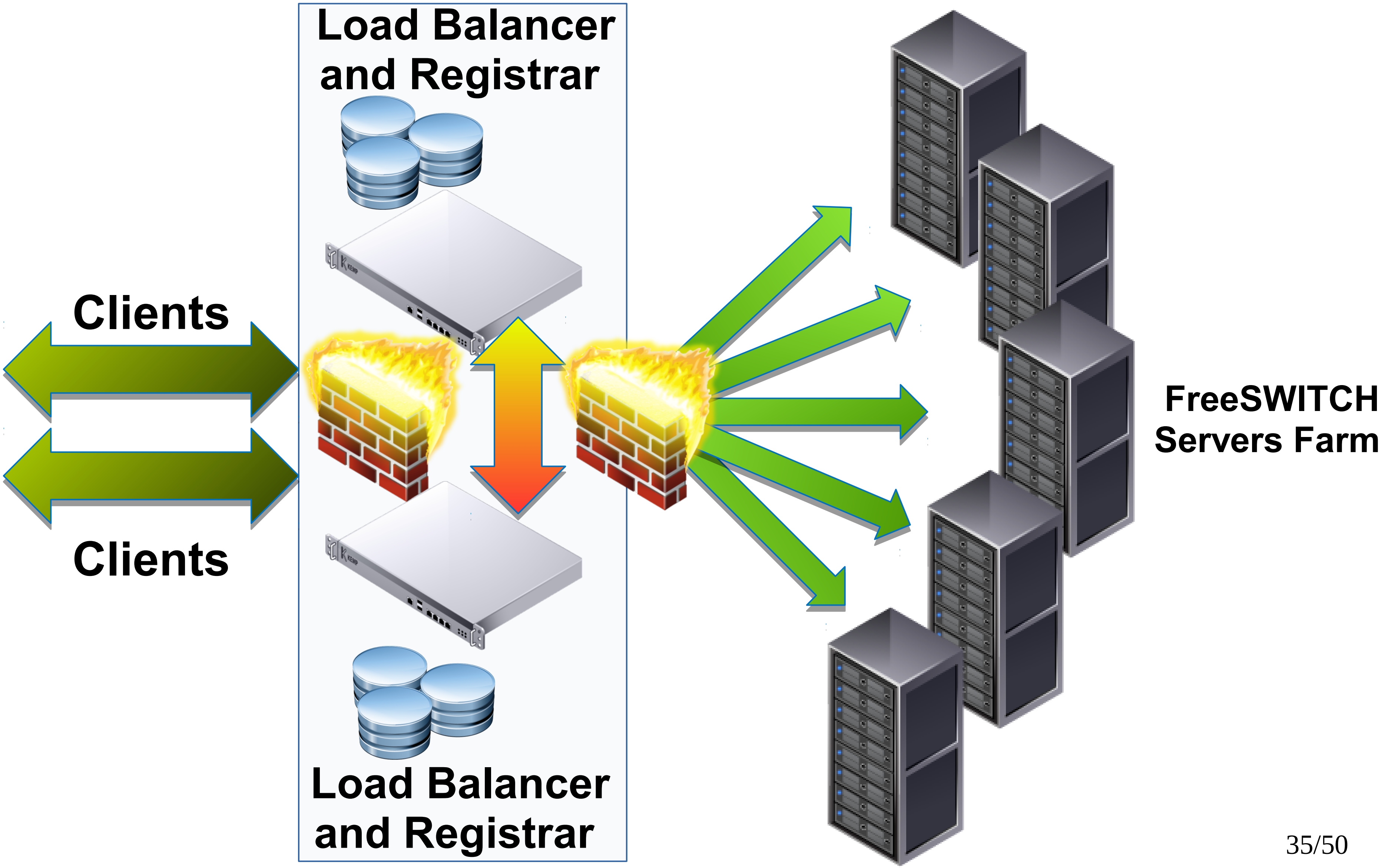
SER Call Distribution: DISPATCHER & LOAD BALANCER

- Both Kamailio and OpenSIPS can use Dispatcher module for relaying (or forwarding – stateless mode) requests to multiple boxes using “static” algorithms (eg: round robin, or weighted)
- For “dynamic” algorithms, that take care of actual number of active calls:
  - You use Dispatcher module in Kamailio (with new “call load distribution” alg)
  - Use Load-Balancer module on OpenSIPS
- Both modules, and all algorithms, are able to “ping” destinations, retry on failed destination, disable the failed box from list, and re-enable when destination is back in order

FS Call distribution: MOD_DISTRIBUTOR

- cd freeswitch_src; make mod_distributor-install
- Simple and effective for alternating between multiple “strings”. Those strings may happen to be the names of sofia gateways :)
- Sofia gateway's subsystem is taking care of failed – back in order gws. Mod_distributor can piggy-back on that, using keywords “gwlist down”
- It can try next destination in case of first destination failure, using “loop” optional argument

Call Distribution



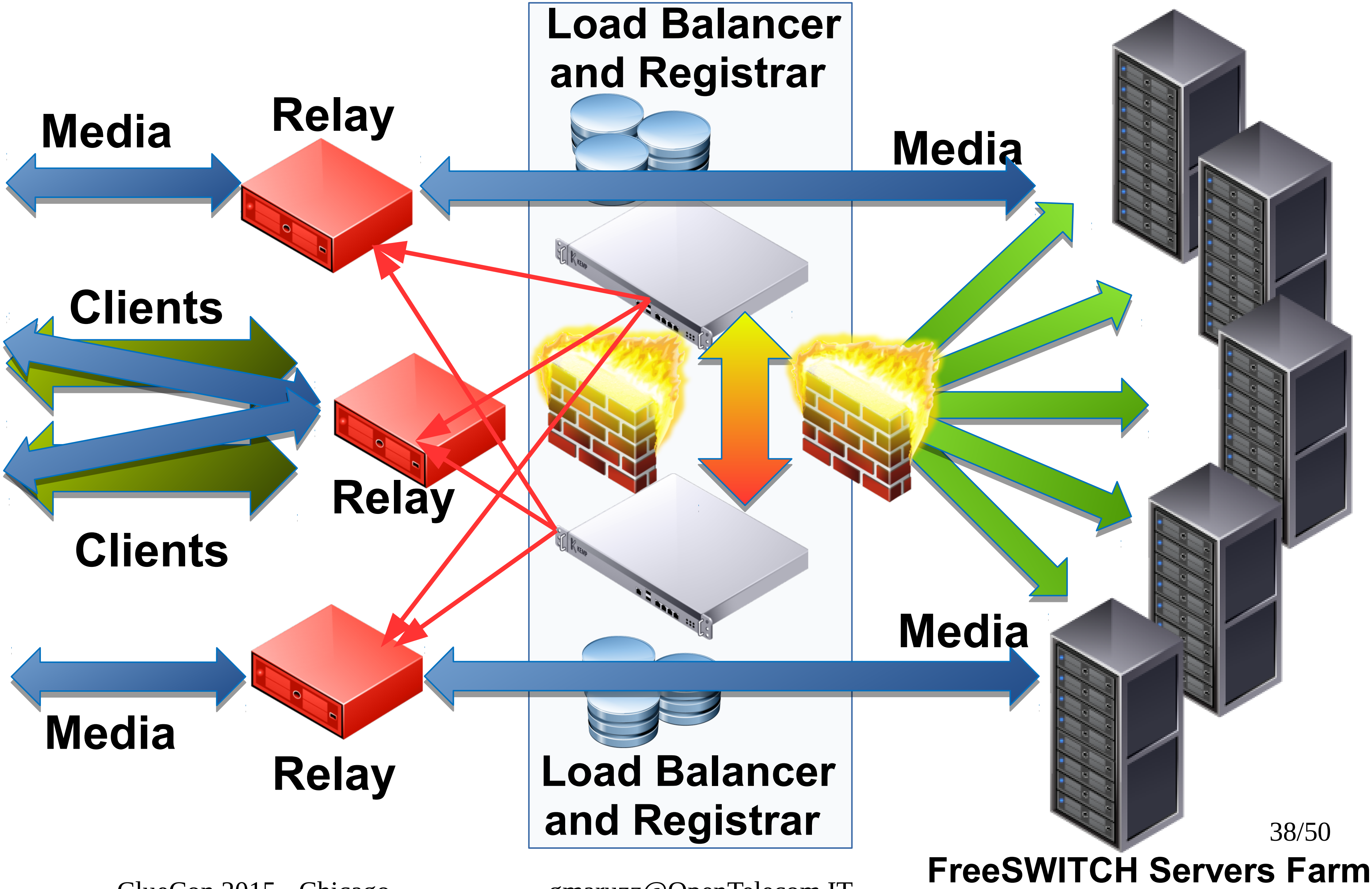
NAT Traversing - Media Relaying

- There are special cases of clients behind NATs that cannot directly sends packets to each other. In those cases ONLY way for them to communicate is via the mediation of a server
- Also, you need to relay media in any case, if you're load balancing servers that are not directly reachable from clients

SER Media Relaying

- Kamailio and OpenSIPS, pure SIP proxies, have nothing to do with media flow, don't touch RTP
- They can modify SIP headers, and SDP bodies, so clients behind restrictive NATs will use a third party as a relay, and they can pass commands to that relay (eg: so it knows which client must be relayed to which)
- Original relay software is “Rtpproxy”
- More recent and advanced (eg: kernel space, etc):
  - Rtpengine (webrtc)
  - Mediaproxy
- All of them can scale indefinitely

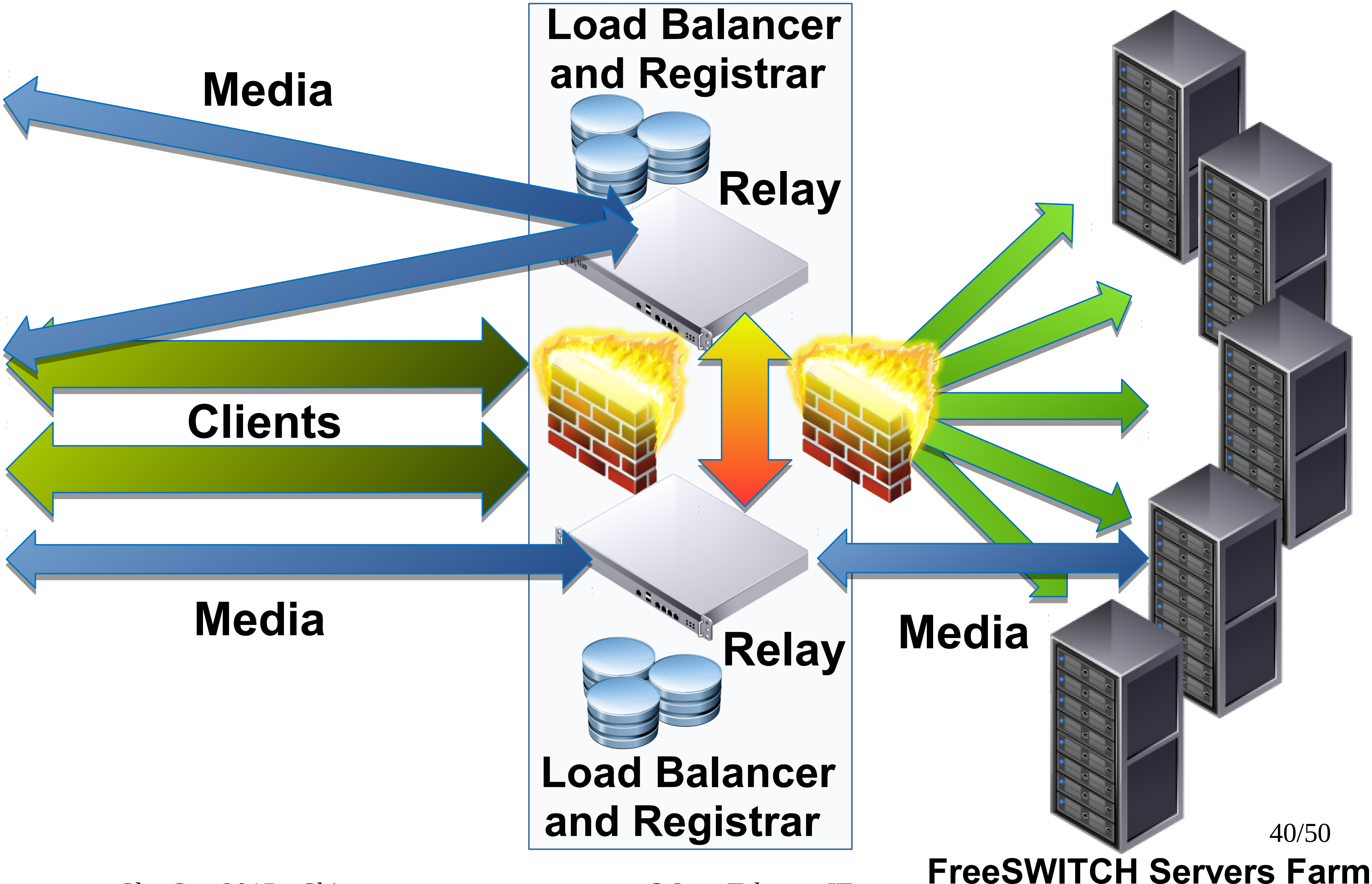
SER Media Relaying



FS Media Relaying (proxy-media)

- FreeSWITCH, being a B2BUA, can definitely be involved in media path
- Default FS operating mode is to be in media path, and actively monitor it (eg: for DTMFs, etc)
- For operating as media relay (less overhead than actively monitoring) you choose “**proxy-media**”, globally on the SIP profile, or for each one call in dialplan
- Also, you can choose to proxy-media a call after it is successfully bridged to destination
- To scale beyond the single machine you must “redirect” the call to another FreeSWITCH server (eg: “bypass-media” on first tier load balancer will redirect calls to second tier “proxy-media” load balancers)

FS Media Relaying



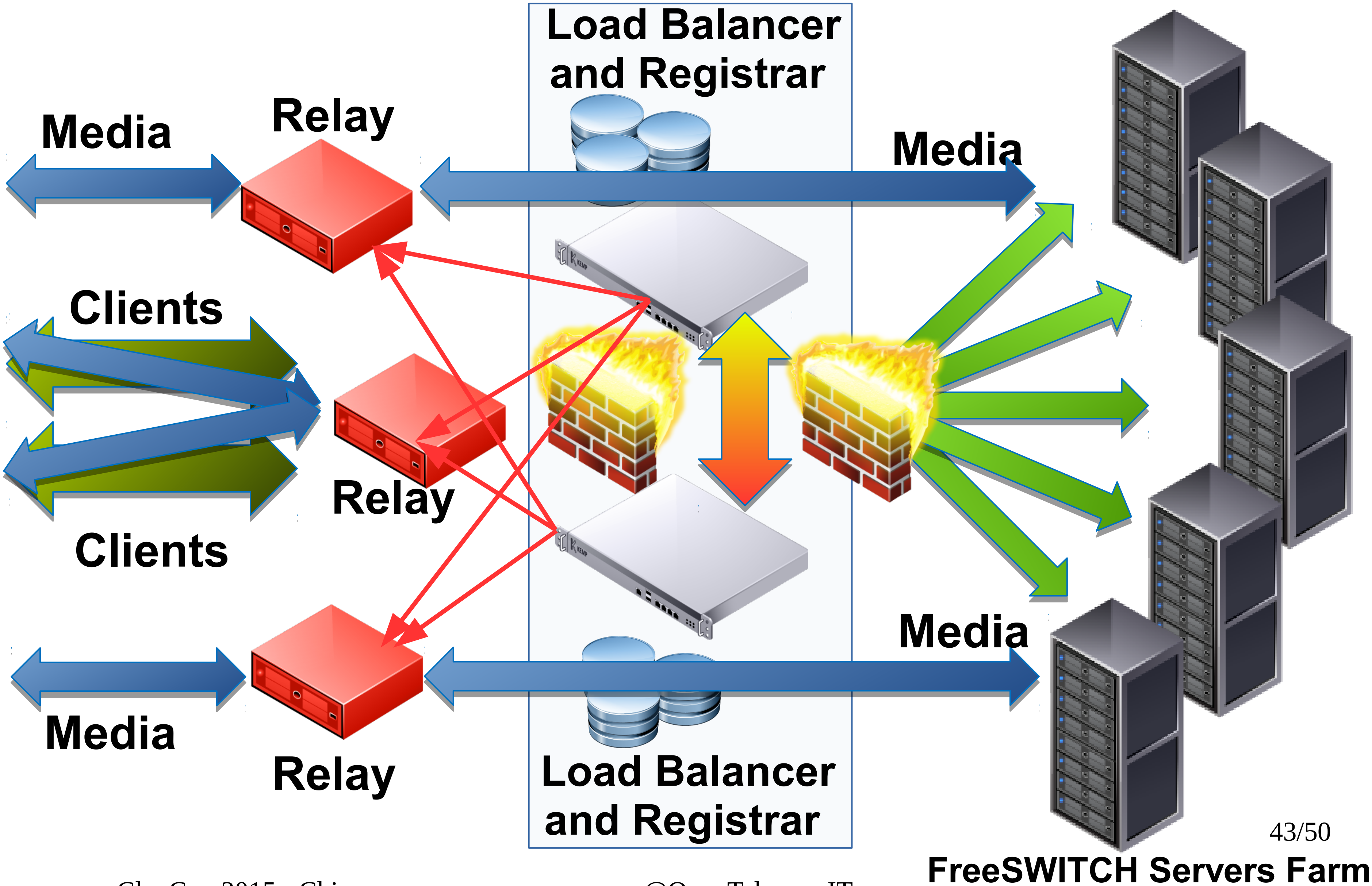
Transport Translation

- SIP can use for signaling:
  - UDP
  - TCP
  - TLS
  - IPv4
  - IPv6
- “SIP” (actually SDP) can use for media:
  - RTP
  - SRTP
  - ZRTP
- Our Load Balancer “SBC”, if properly configured, will have no problem to translate signaling transports, both Kamailio, OpenSIPS and FreeSWITCH can have many “profiles” (defined by IPaddress:port) on same instance, each one “talking” one transport

SER Media Transport Translation

- Rtpengine can be used with Kamailio to translate media trasport between rtp and srtp
- OpenSIPS has beta support for Rtpengine
- Not yet support to relay ZRTP via Kamailio or OpenSIPS

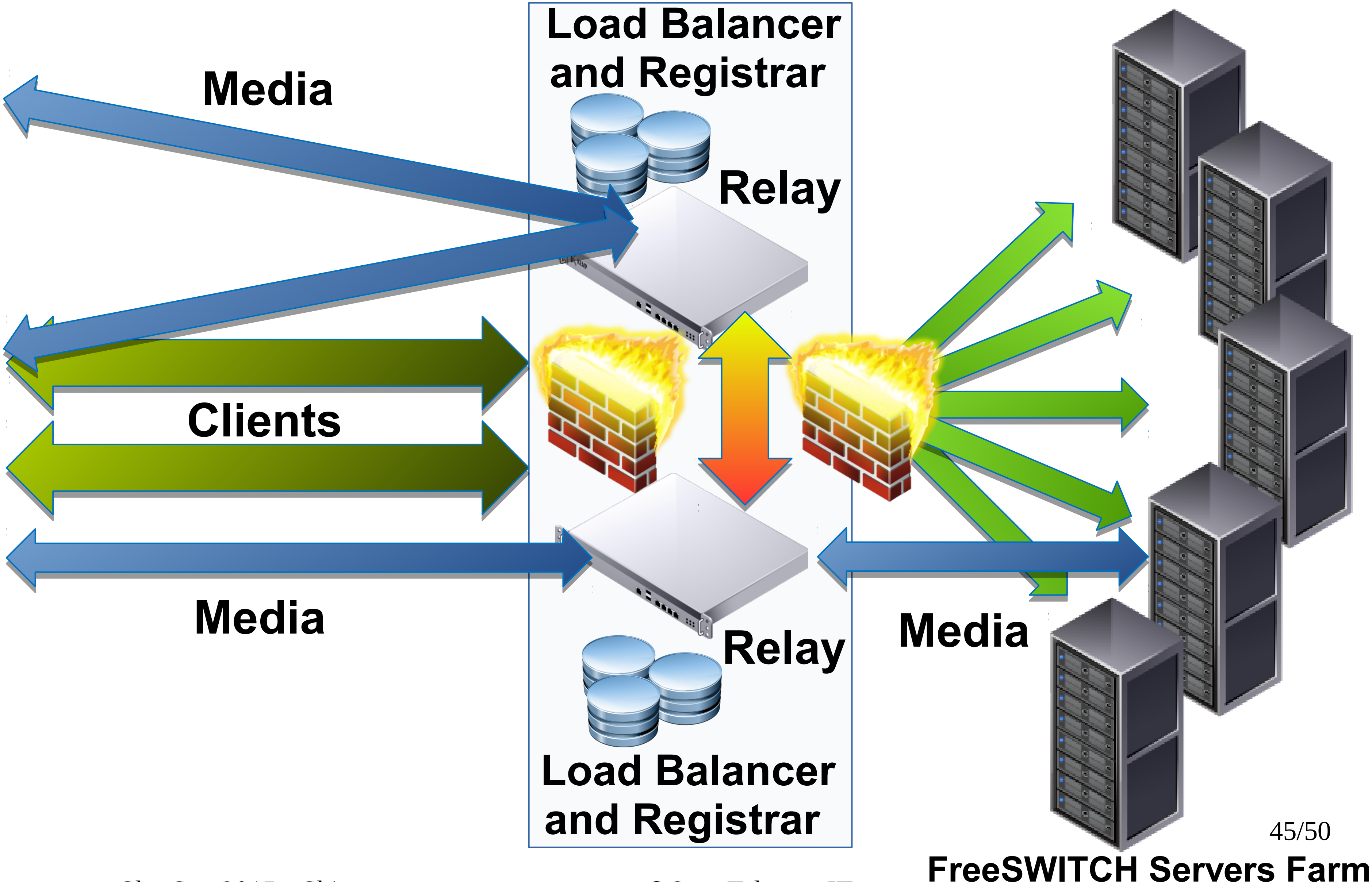
SER Media Transport Translation



FS Media Transport Translation

- Being a B2BUA, FreeSWITCH accepts an incoming “A” call leg, generates an outbound “B” leg, and joins (“bridge”) them
- A and B legs are completely unrelated entities
- They can obviously use different media transports

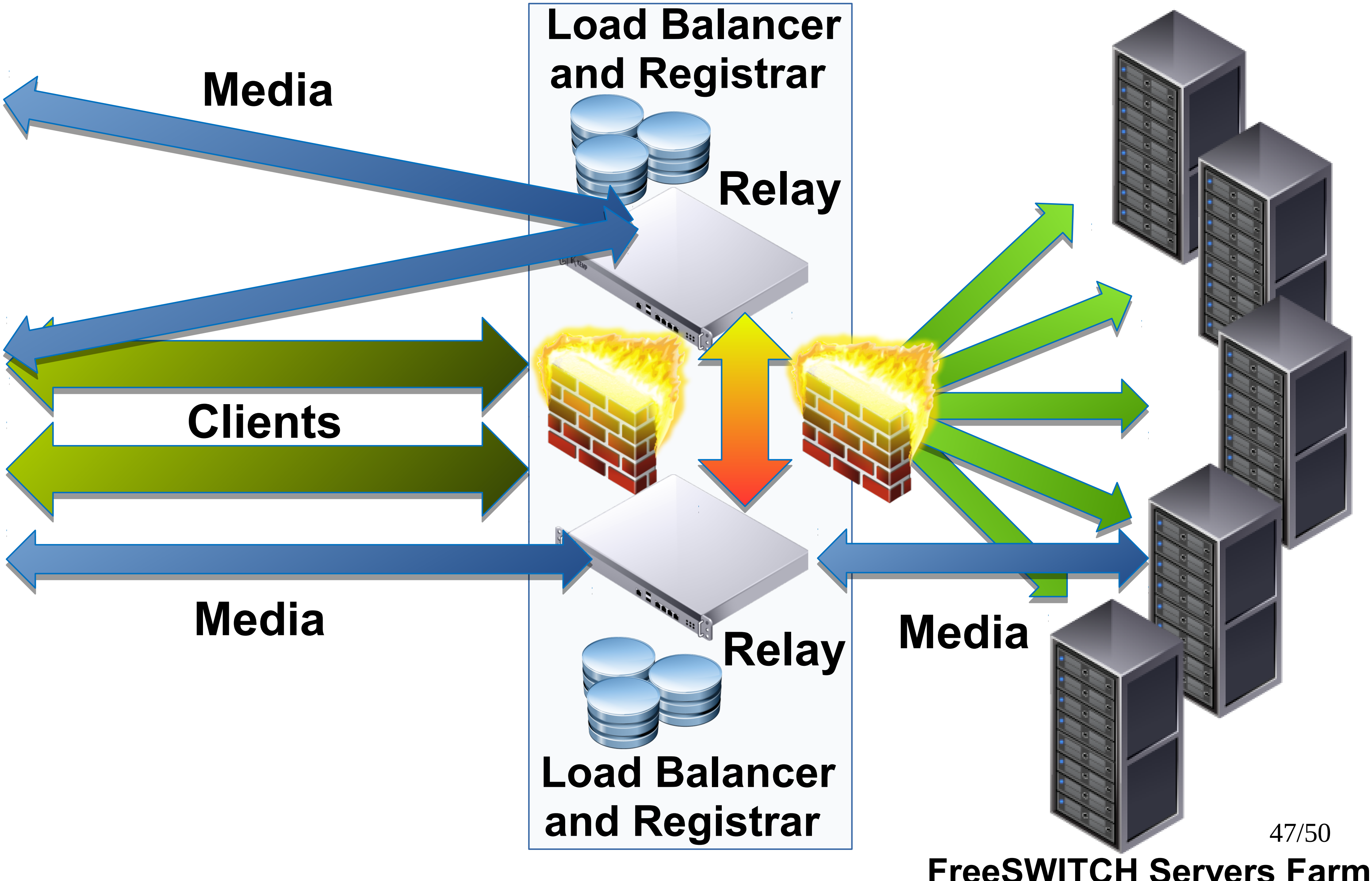
FS Transport Translation



Media Transcoding (FS)

- Media High Performance Mixing, Muxing and Transcoding (**Audio and Video**) is one of FreeSWITCH strengths
- FreeSWITCH supports High Definition (HD) audio, up to 48khz, dozen of codecs, g711, g723, g729, OPUS, Siren, Ilbc, Speex, Codec-2
- FreeSWITCH can encode, decode and mux V8 and H264 video streams, up to 2420 pixel wide

FS Media Transcoding



WebRTC !

- STUN
- TURN
- ICE Candidates
- Much more ports needed!
- Corporate Firewalls (only ports 80 and 443)
- PSTN interconnect: Audio Codecs are different! (G711 vs OPUS)
- Transports are differents! (SRTP-DTLS vs UDP-TCP-TLS)

Special cases

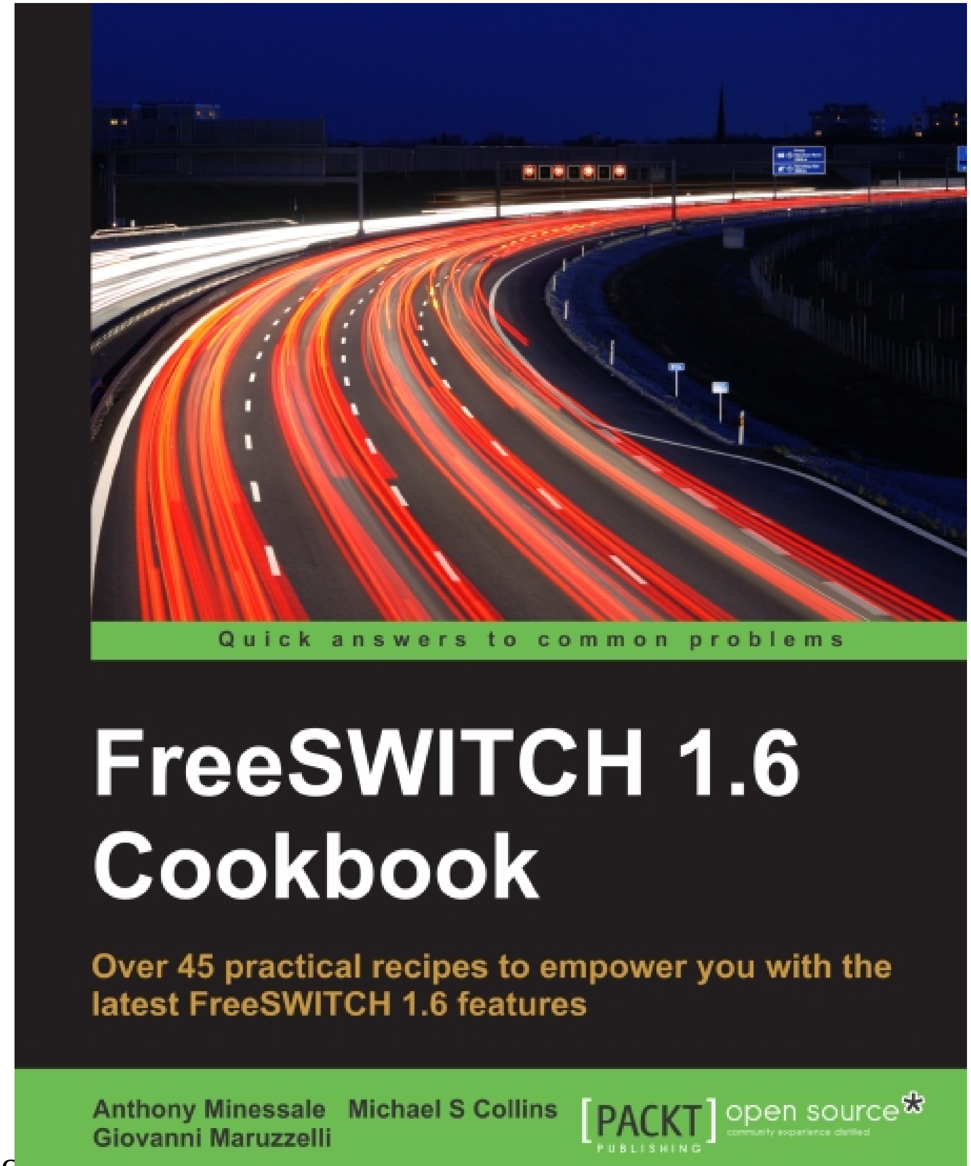
- Load Balancing is predicated on a server farm of equivalent and equipollent (eg: interchangeable) servers

- There are cases for which this is not true:
  - Conferences
  - Call queues
  - Call centers

- **ANSWER IS: Custom code!**

www.packtpub.com

25% Discount Code: **CLUECON25**



Thank You

QUESTIONS ?

Giovanni Maruzzelli
gmaruzz@OpenTelecom.IT

OpenTelecom.IT
opensource telecommunication