



Server Speak

Mike MacIsaac, Sine Nomine Associates

mmacisaac@sinenomine.net

VM Workshop, Richmond VA

June 22nd 2024



Abstract

We should be able to ask our CMDB the status of our systems. This presentation will describe a prototype that does just that. You say "How many of my servers are down?", it pings every server and speaks status to you. Tie this app into mobile devices, and the CIO/CTO types love it.

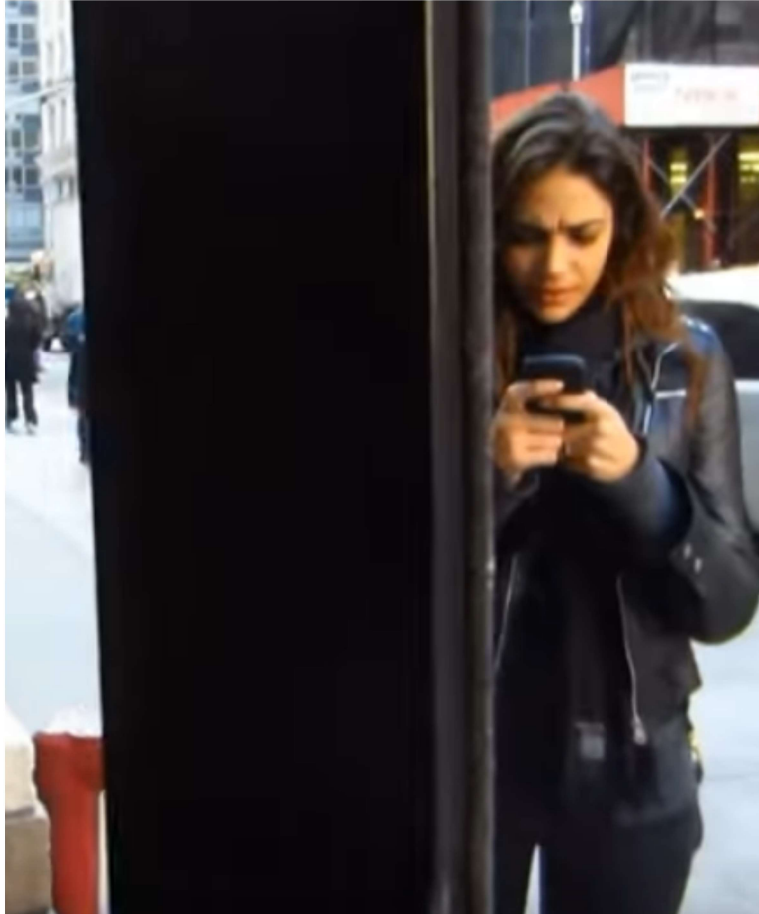


Bio

- Past jobs
 - IBM (27 yrs)
 - Innovation (2 yrs)
 - ADP (8 yrs)
- Current job: Sine Nomine Associates
- Anniversaries
 - Mainframe/Ford Mustang (60 years)
 - VM logon (40 years)
 - zLinux login (25 years)
 - Personal voice assistants (5 years)



Can you walk and text at the same ti



Will we think back: “Remember when we had to ***TYPE*** on our ***PHONES?***”

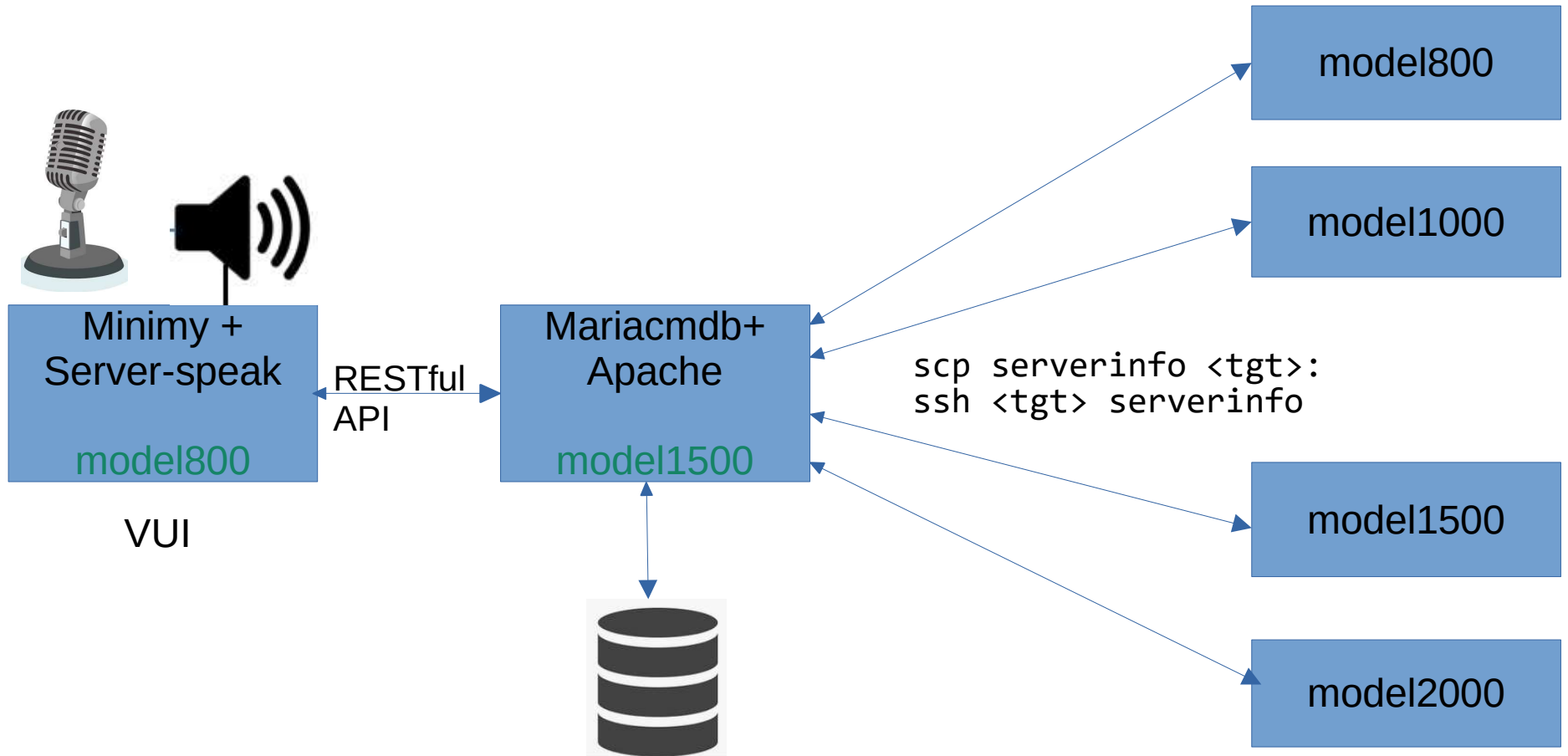


Server Speak overview

- Get IT status by voice – Components:
 - **mariaacmdb** – Simple Configuration Management Databases
 - Uses **mariadb** relational database (**50 years for RDBs**)
 - Line command **mariaacmdb.py** – init, query, add, delete
 - RESTful API **restapi.py** – query w/Apache (R/O)
 - Bash script **serverinfo** returns basic server data
 - **Minimy** – Personal voice assistant stack
 - **server-speak** – IT “skill”
 - Requires microphone and speaker(s)
 - **ssh/sshd** – Key-based authentication avoids passwords
 - **Ansible** – Possible alternative



Server Speak block diagram



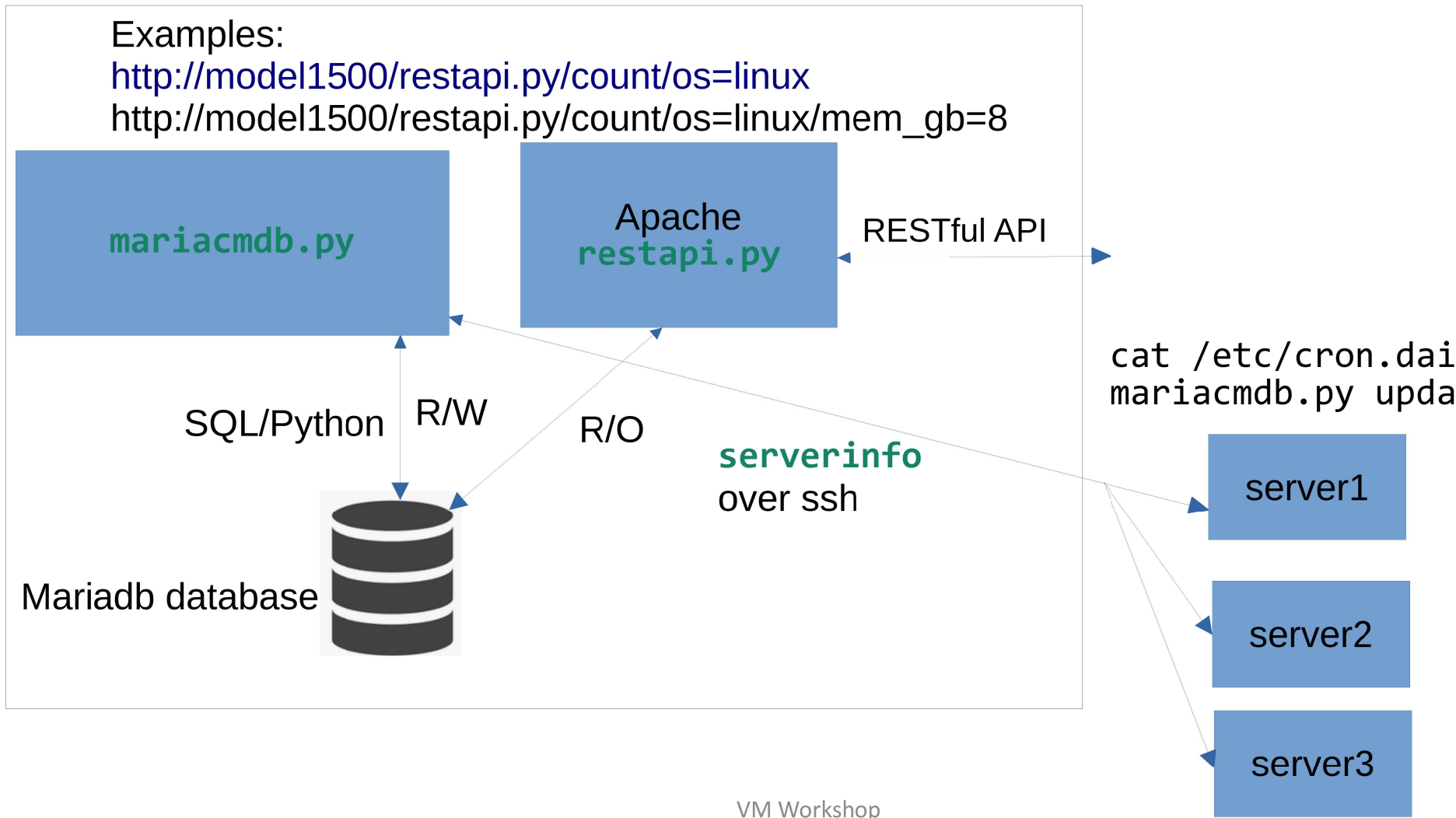


Let's try a demo

- **The (canned) demo:**
 - **Runs from model800 (minimy)**
 - **Monitors RESTful API from model1500 (mariacmdb)**
- **List Environment - `lsenv`**
- **Count servers**
 - **Computer, how many servers do we have?**
 - **Computer, how many Linux servers do we have?**
 - ...



mariacmdb block diagram

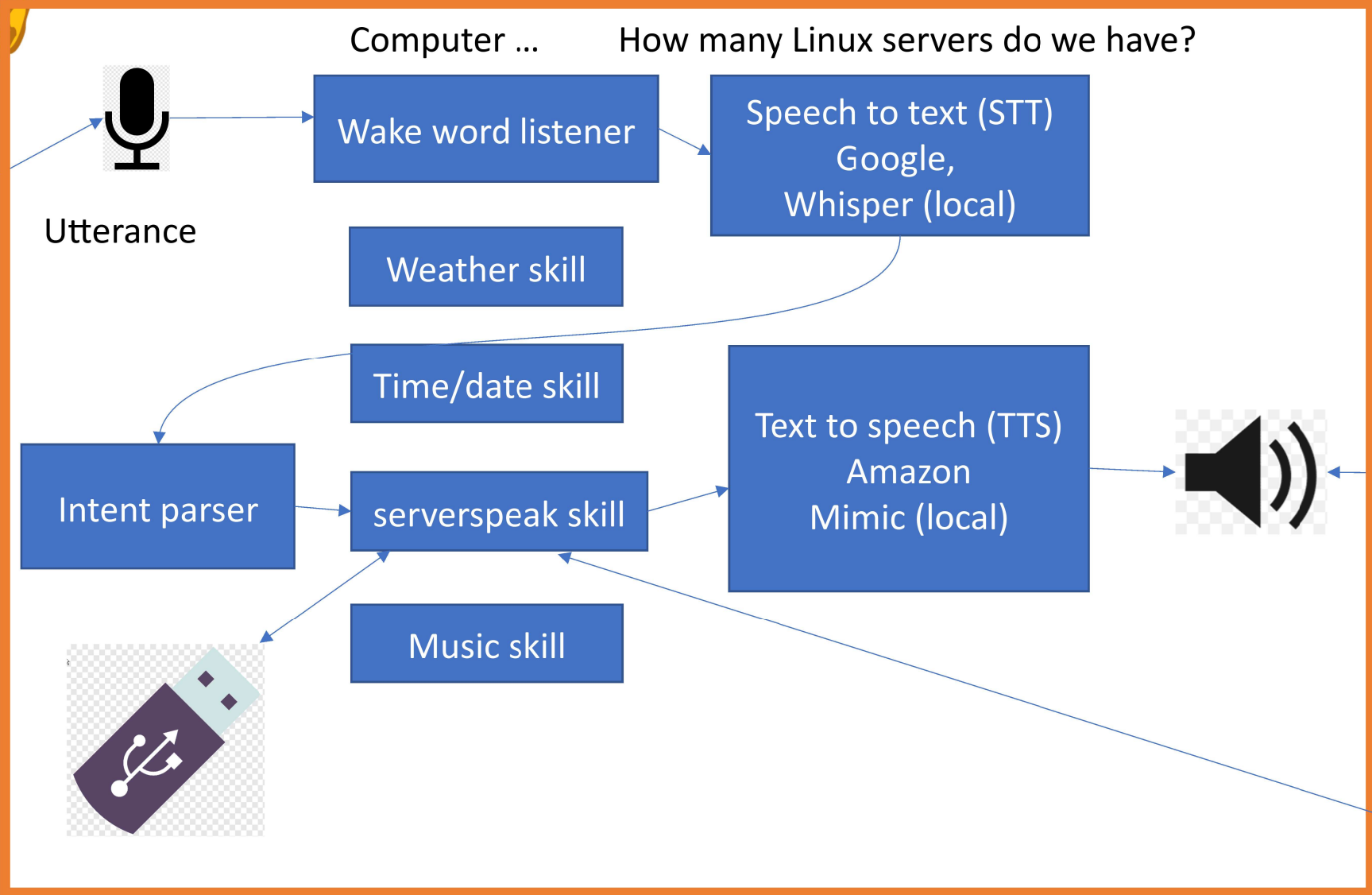




Minimy block diagram

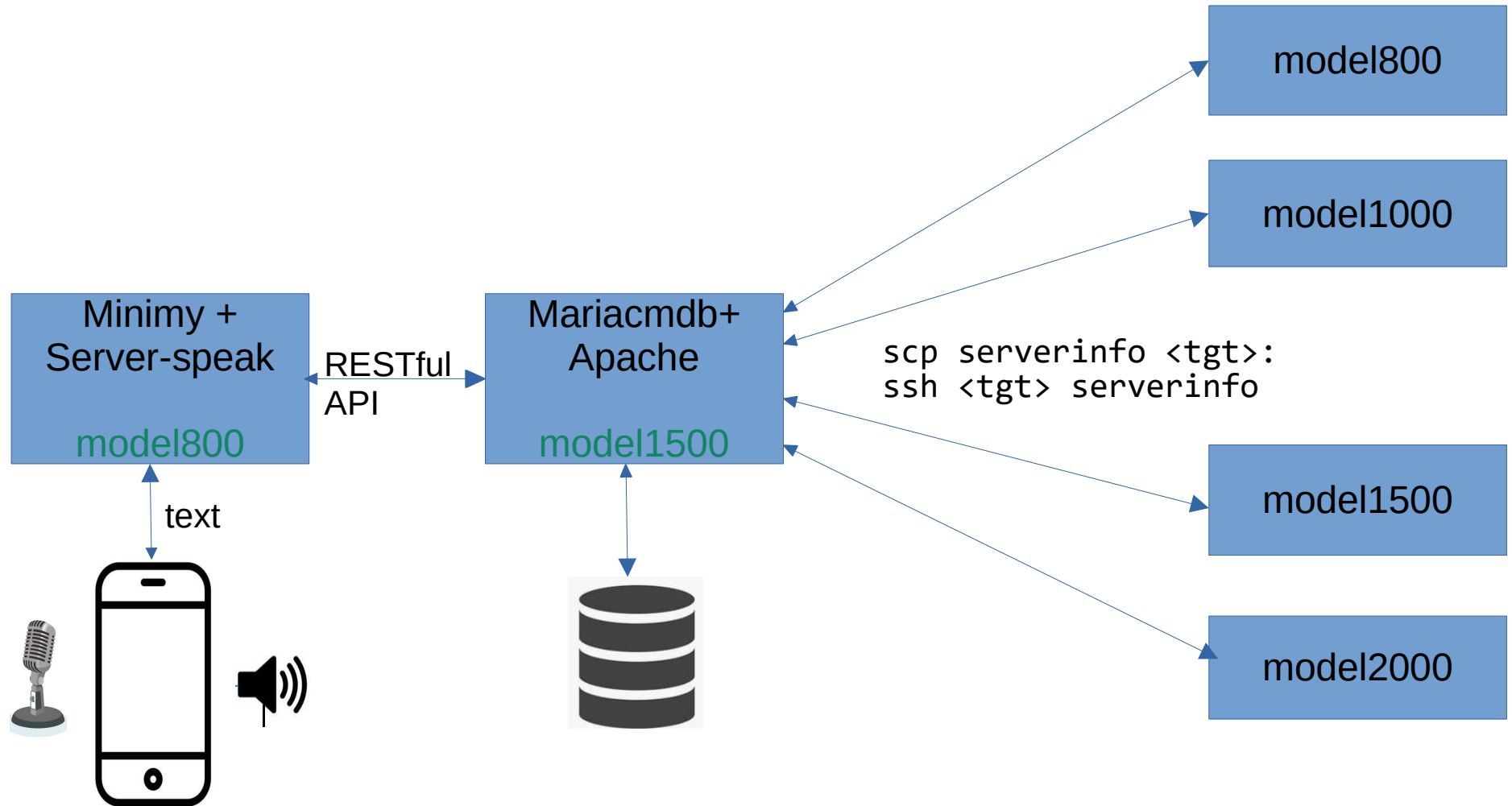


Utterance



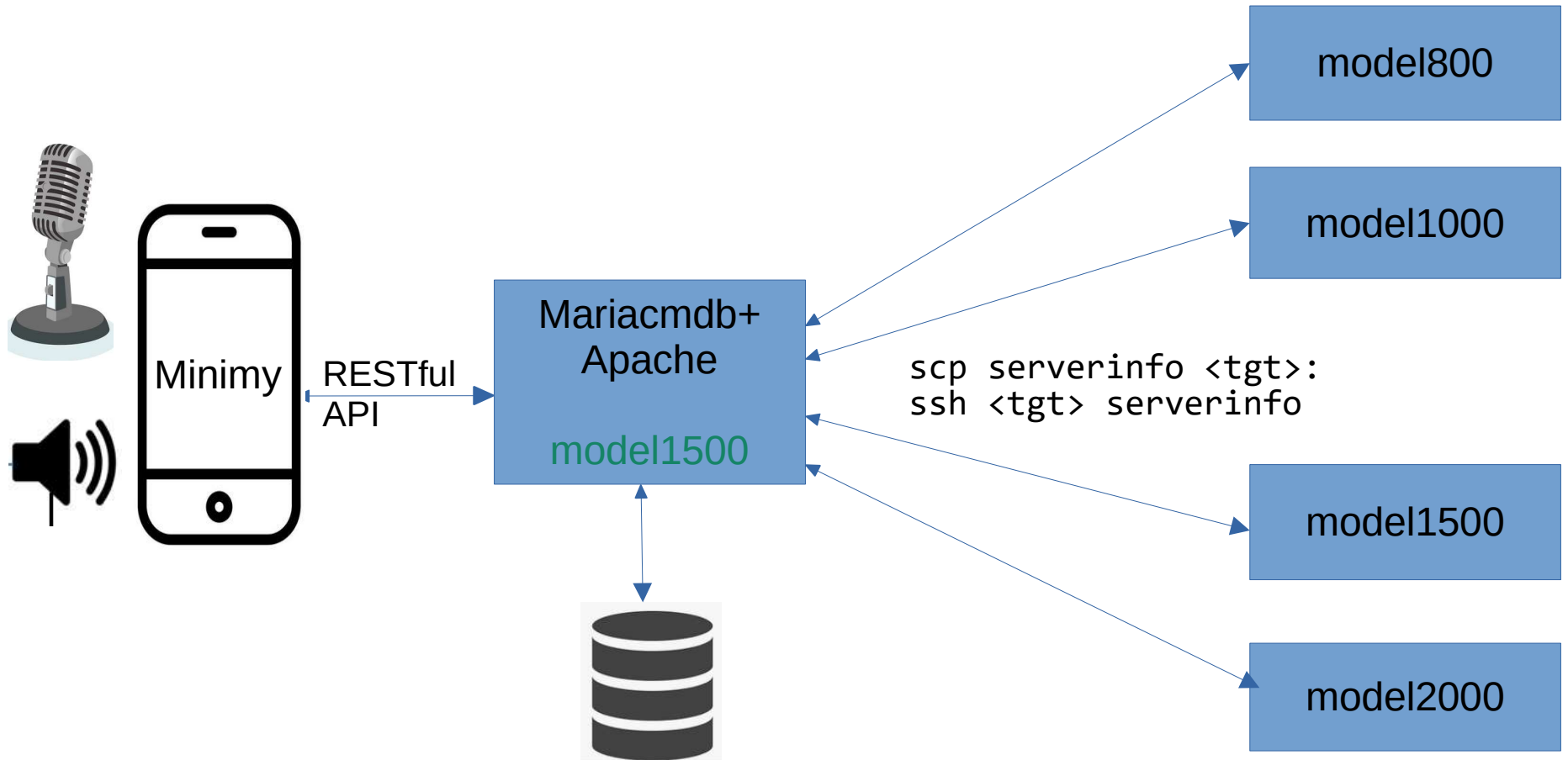


Adding mobile devices



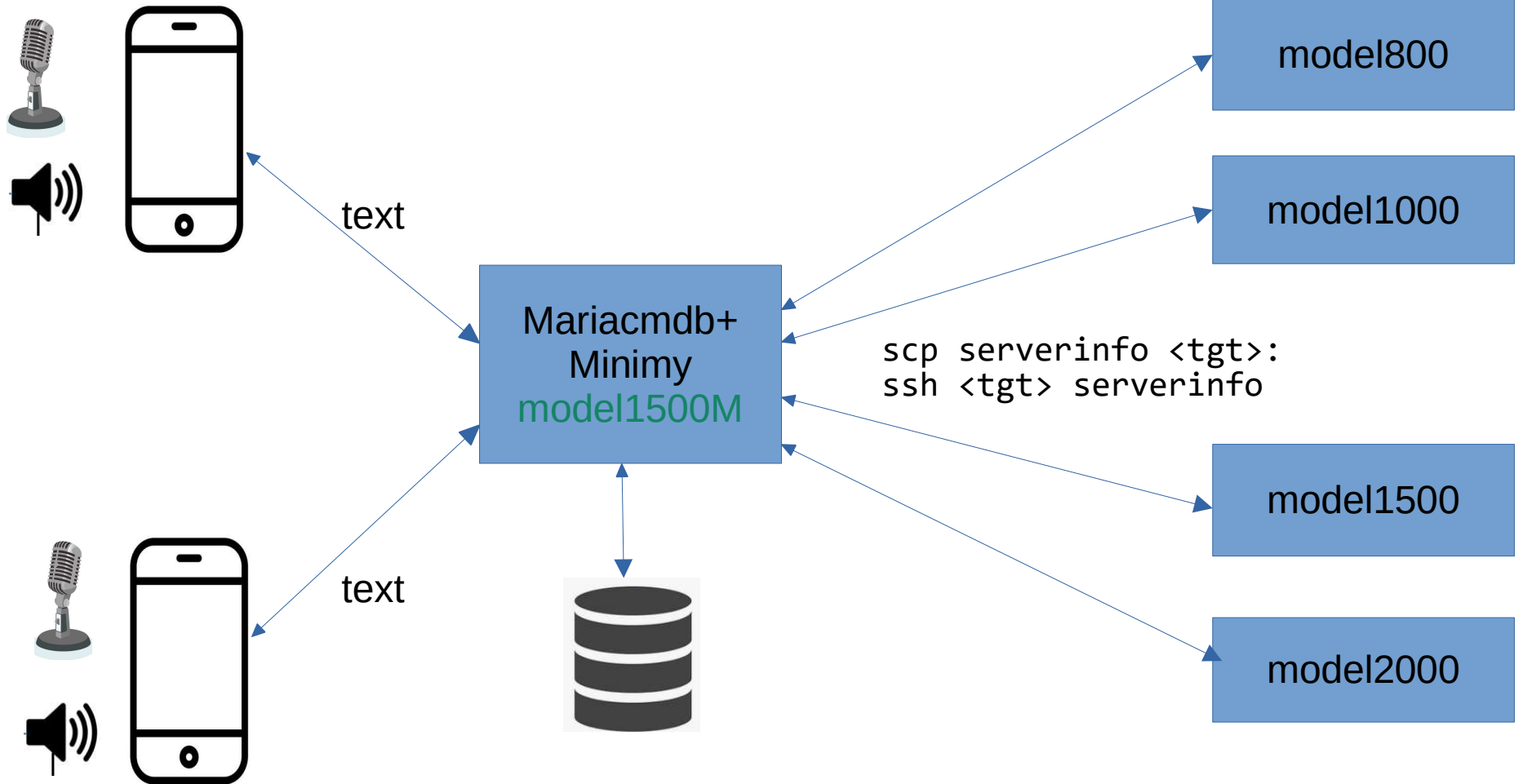


A simpler model?





YAM?





Mariacmdb table servers

Field	Type	Null	Key	Default	Extra
host_name	varchar(255)	NO	PRI	NULL	
ip_addr	varchar(20)	YES		NULL	
cpus	int(11)	YES		NULL	
mem_gb	int(11)	YES		NULL	
arch	varchar(50)	YES		NULL	
os	varchar(100)	YES		NULL	
os_ver	varchar(50)	YES		NULL	
kernel	varchar(100)	YES		NULL	
rootfs	int(11)	YES		NULL	
created_at	timestamp	NO		current_timestamp()	

VM Workshop



Mariacmdb new columns

- **arch_com** – commonly used architecture
- **app**
- **group**
- **last_ping**
- **owner**
- **???**



Serverinfo script (condensed)

```
hostName=`hostname`
IPAddr=`ip route get 8.8.8.8 | sed -n '/src/{s/.*/src *\[^\ ]*\).*\/\1/p;q}'`
...
numCPUs=`grep ^processor /proc/cpuinfo | wc -l`
memKB=`cat /proc/meminfo | head -1 | awk '{print $2}'`
cmd="echo \"scale=1; $memKB/1048576\" | bc" # KB to GB
memGB=`eval $cmd` # do the math
memGB=`printf "%.0f" $memGB` # round to nearest int
arch=`arch` # hardware platform
os=`uname -s` # kernel name
distro=`head -1 /etc/os-release | sed -e 's/PRETTY_NAME=//g' -e 's/"//g'`
if [ ${#distro} = 0 ]; then # empty var
    distro="unknown"
fi
kernel=`uname -r -v` # kernel release and version"
rootfsFull=`df -h / | tail -1 | awk '{print $5}' | sed 's/%//g'`
echo "$hostName,$IPAddr,$numCPUs,$memGB,$arch,$os,$distro,$kernel,$rootfsFull"
```



Mariacmdb command line

```
mariacmdb.py -h
```

```
usage: mariacmdb.py [-h] [-v] [-c]
```

```
    [--column COLUMN] [--value VALUE] [--pattern PATTERN] [--server SERVER] subcommand
```

```
mariacmdb - A simple Configuration Management Database
```

```
positional arguments:
```

```
    subcommand          Can be 'add', 'describe', 'initialize', 'query', 'remove' or 'update'
```

```
options:
```

```
-h, --help            show this help message and exit
```

```
-v, --verbose         increase output verbosity
```

```
-C, --copyscript      copy script 'serverinfo' to target server before add
```

```
-c COLUMN, --column COLUMN    column name to search
```

```
-p PATTERN, --pattern PATTERN  pattern to search for
```

```
-s SERVER, --server SERVER    server to add or remove
```




mariacmdb.py subcommands

- **mariacmdb.py** subcommands:
 - **add** Add server to be managed
 - Update if already exists
 - **describe** Show the metadata of the servers table
 - **initialize** Create the table **servers**
 - **query** Show specified rows
 - **remove** Remove a managed server
 - **update** Update all rows



mariacmdb.py query

mariacmdb.py query

```
model1000,192.168.1.229,4,4,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6.2712 #1 SMP PREEMPT Debian 1:6.6.28-1+rpt1 (2024-04-22),25,2024-05-21 09:23:32
```

```
model1500,192.168.1.147,4,4,aarch64,Linux,Ubuntu 22.04.4 LTS,5.15.0-1055-rasbp SMP PREEMPT Sat May 4 03:52:40 UTC 2024,24,2024-05-21 09:24:02
```

```
model2000,192.168.1.102,4,8,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6.2712 #1 SMP PREEMPT Debian 1:6.6.28-1+rpt1 (2024-04-22),30,2024-05-22 11:19:30
```

```
model800,192.168.1.35,4,4,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6.2712 #1 SMP PREEMPT Debian 1:6.6.28-1+rpt1 (2024-04-22),30,2024-05-22 11:15:30
```



mariacmdb.py add and remove

```
$ mariacmdb.py remove -s model1800
```

```
__main__ : INFO delete_row(): deleted row for server model1800
```

```
$ mariacmdb.py query
```

```
model1000,192.168.1.229,4,4,aarch64,Linux,Ubuntu 22.04.4 LTS,5.15.0-1055-ras
```

```
model1500,192.168.1.147,4,4,aarch64,Linux,Ubuntu 22.04.4 LTS,5.15.0-1055-ras
```

```
model2000,192.168.1.102,4,8,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6
```

```
$ mariacmdb.py add -s model1800
```

```
__main__ : INFO replace_row(): replaced row for server model1800
```

```
$ mariacmdb.py query
```

```
model1000,192.168.1.229,4,4,aarch64,Linux,Ubuntu 22.04.4 LTS,5.15.0-1055-ras
```

```
model1500,192.168.1.147,4,4,aarch64,Linux,Ubuntu 22.04.4 LTS,5.15.0-1055-ras
```

```
model2000,192.168.1.102,4,8,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6
```

```
model1800,192.168.1.35,4,4,aarch64,Linux,Debian GNU/Linux 12 (bookworm),6.6.2
```



Demo 2 – ping servers

- **Ping servers**
 - **Computer, how many of my servers are up?**
 - **Reboot one**
 - **Computer, how many of my servers are down?**



RESTful API

- **Examples using curl command:**

```
curl http://model1500/restapi.py/count
```

```
{"num_servers": 4}
```

```
curl http://model1500/restapi.py/count/mem_gb=4
```

```
{"num_servers": 3}
```

```
curl http://model1500/restapi.py/ping
```

```
{"up_servers": 4, "num_servers": 4}
```

- **Sending data to dialog file:**

```
cat up_servers.dialog
```

```
{up_servers} of {num_servers} ping
```



Mariacmdb source files

- **Lines of Python and bash code:**

```
359 /usr/local/sbin/mariacmdb.py
```

```
148 /srv/www/mariacmdb/restapi.py
```

```
37 /home/pi/serverinfo
```

```
544 total
```



Sources

- Where do I get the code?
 - Mariacmdb - <https://github.com/mike99mac/mariacmdb>
 - Minimy - <https://github.com/mike99mac/minimy-mike99mac>
 - Bonus: music by voice
 - Server-speak skill: Still under development
- Questions installing/using?
 - Email: mmacisaac@sinenomine.net



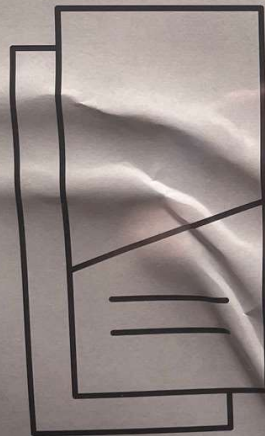
On the lighter side...



PLEASE HELP KEEP THE ALC CLEAN AND DAMAGE FREE

DO YOU HAVE FLYERS TO HANG UP??

MAKE SURE TO KEEP THEM OFF THE WALLS AND
ON THE BULLETIN BOARDS :)



DON'T DO IT!

Any Flyers Hung On The
Walls Will Be Taken
Down.



Boomboxes



mo

My

Ra
DA
fan

VM Workshop



3 Boomboxes

