facebook

The osquery File Carver

Nick Anderson

Security Engineer

C:\> Get-Host

- Nick Anderson
  - Security Engineer at Facebook
- thor@fb.com
  - Super legit, not an alias
- Github - github.com/muffins
- Twitter - twitter.com/poppyseedplehzr
- Slack - thor

A quick story

Analyst

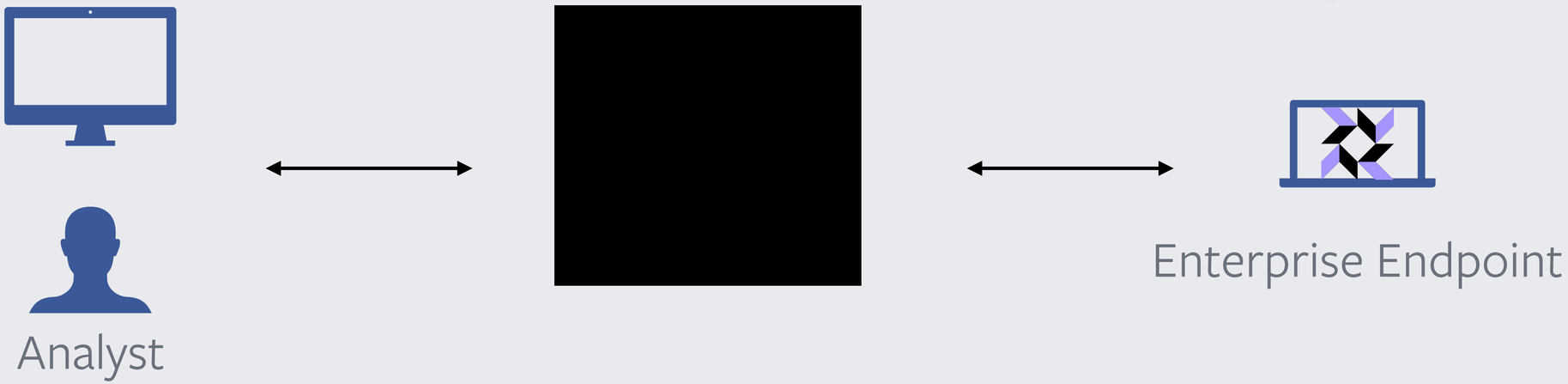
Enterprise Endpoint

/tmp/evil

Analyst

Enterprise Endpoint

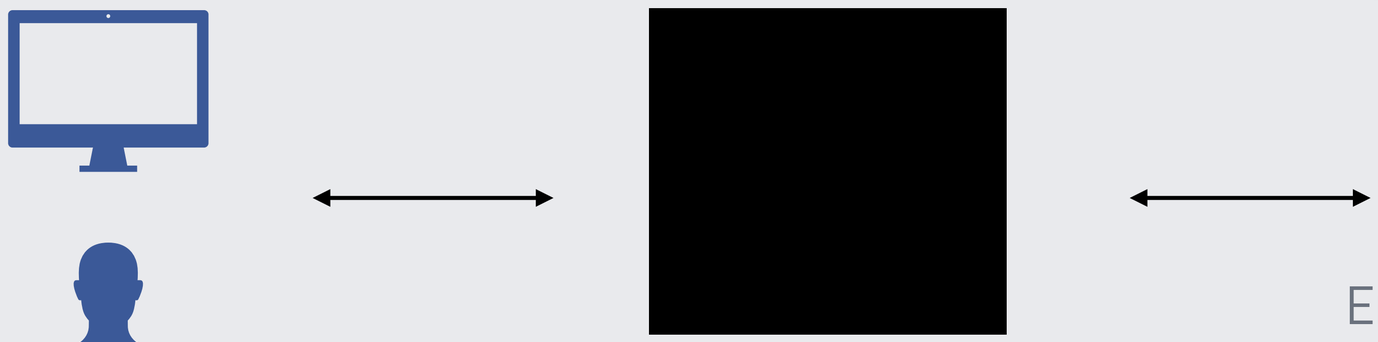
/tmp/evil



Analyst

Enterprise Endpoint

/tmp/evil



Analyst

Enterprise Endpoint

```
#!/bin/bash
nc -l -p 1337
```

/tmp/evil


```
#!/bin/bash
nc -l -p 1337
```

What is osquery?

What is osquery?

- FOSS host based IDS

What is osquery?

- FOSS host based IDS
- Cross-platform

What is osquery?

- FOSS host based IDS
- Cross-platform
- Abstracts OS as SQLite tables

What is osquery?

- FOSS host based IDS
- Cross-platform
- Abstracts OS as SQLite tables
- Performant

What is osquery?

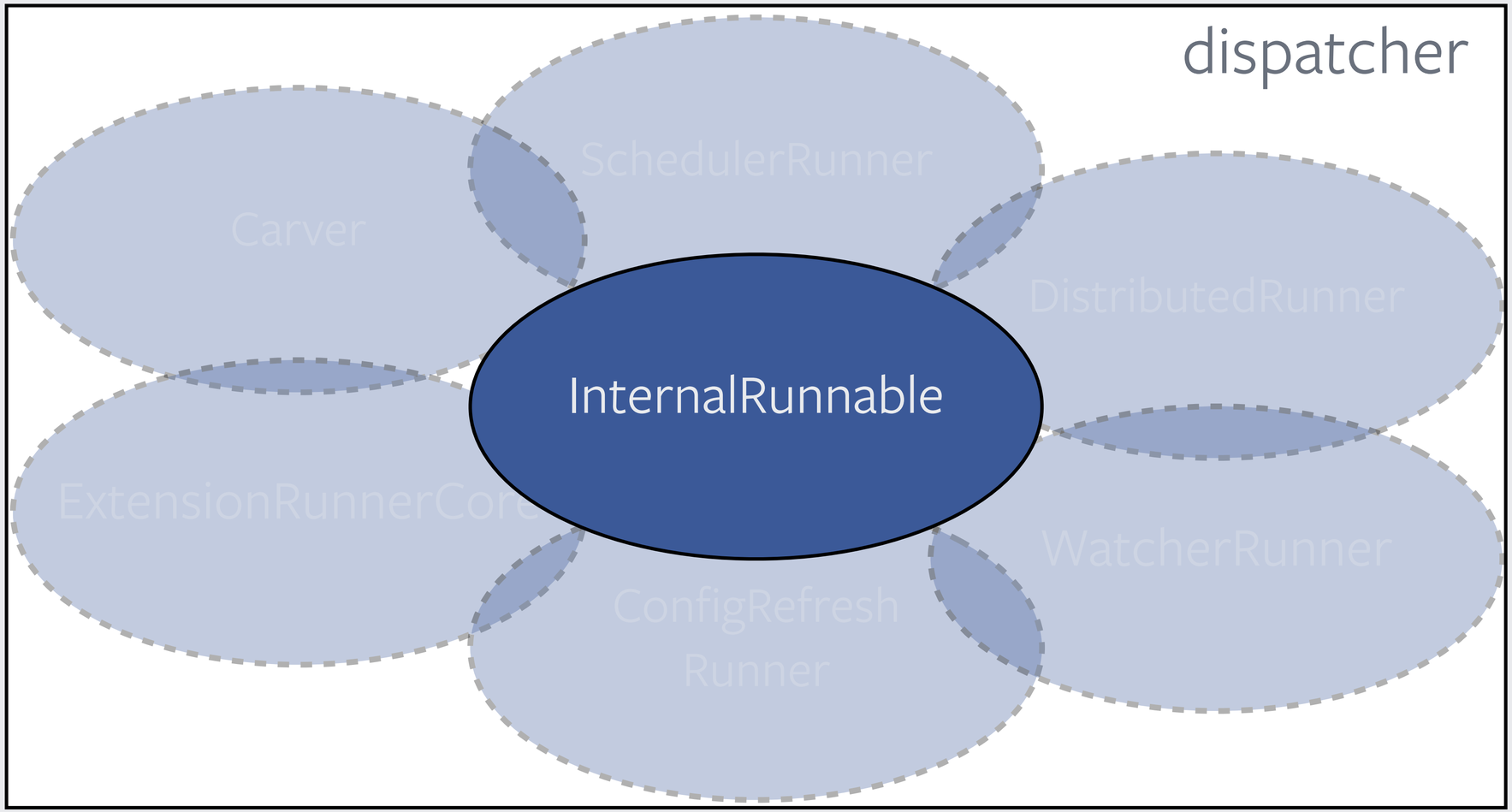
- FOSS host based IDS
- Cross-platform
- Abstracts OS as SQLite tables
- Performant
- Extensible

What is osquery?

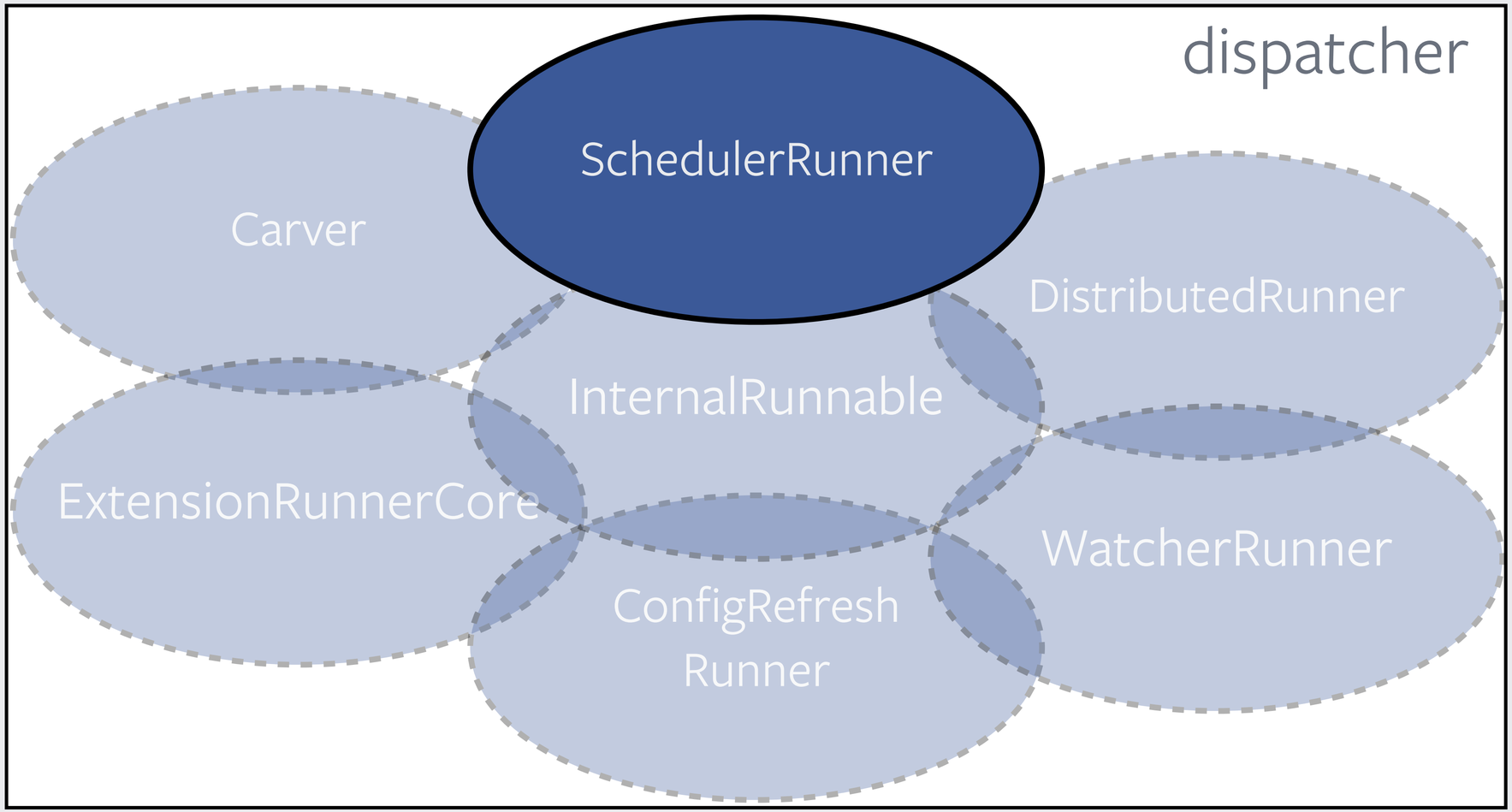
- FOSS host based IDS
- Cross-platform
- Abstracts OS as SQLite tables
- Performant
- Extensible
- https://osquery.io

osquery architecture

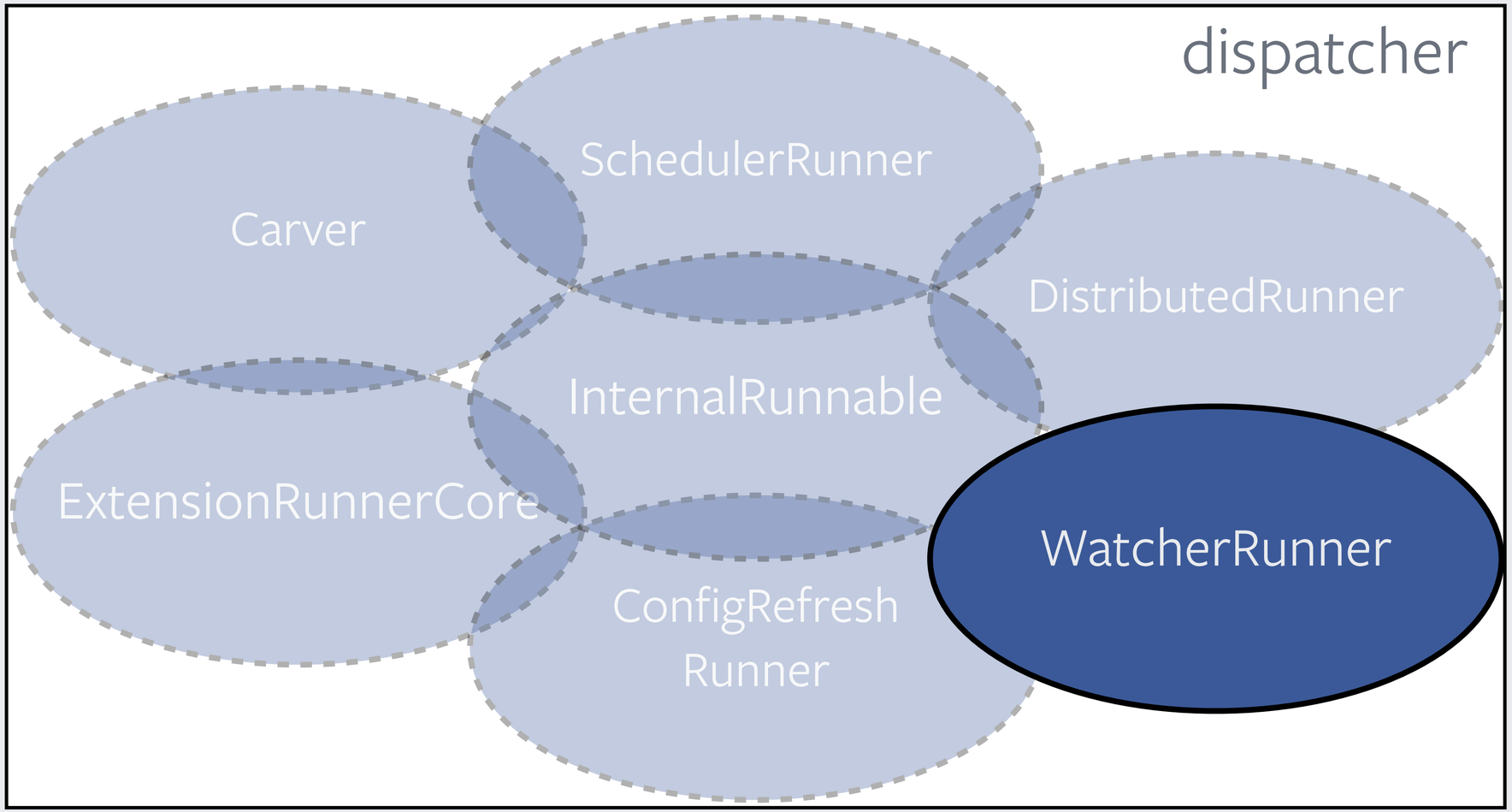
How does osquery work?



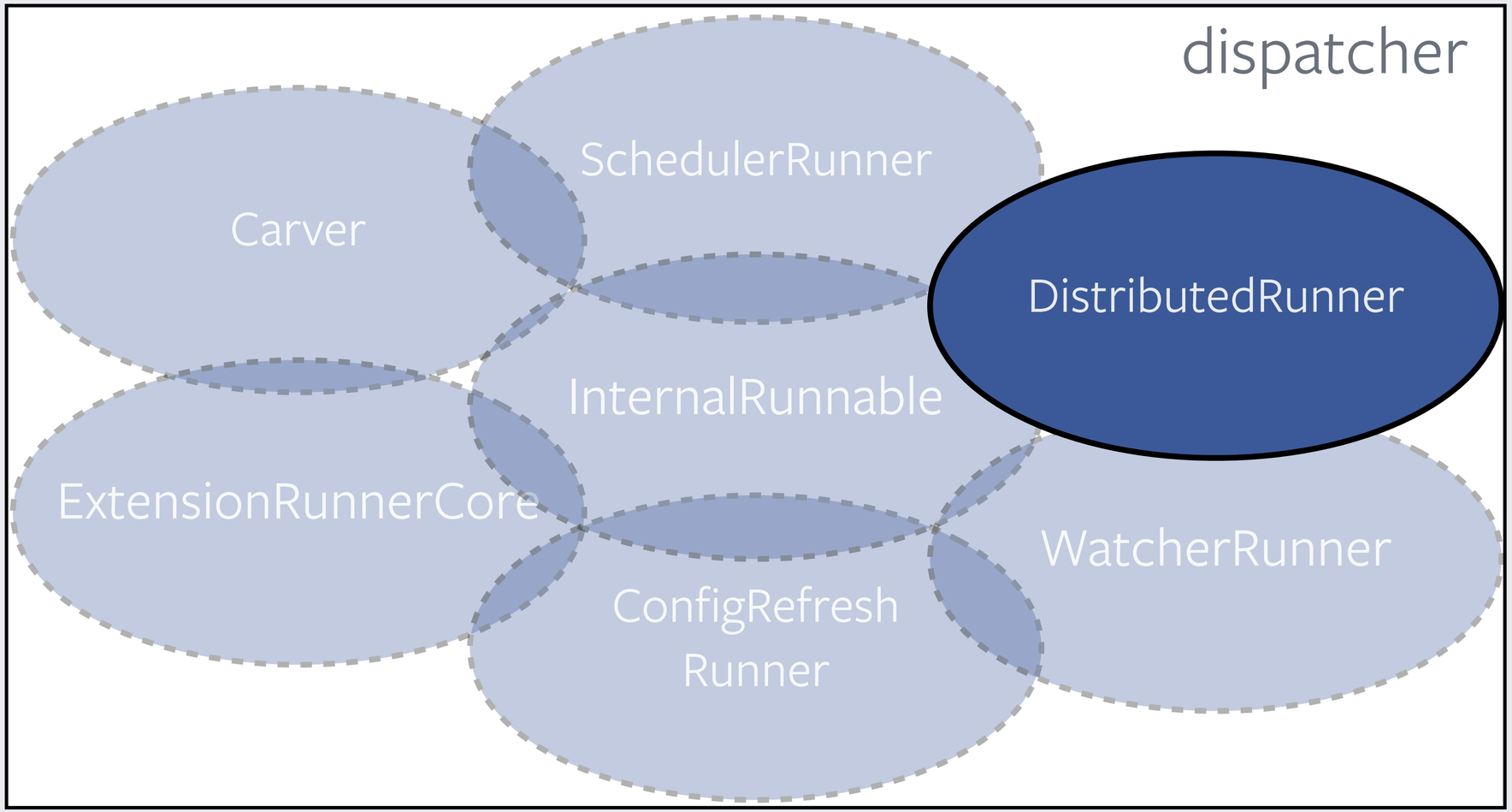
How does osquery work?



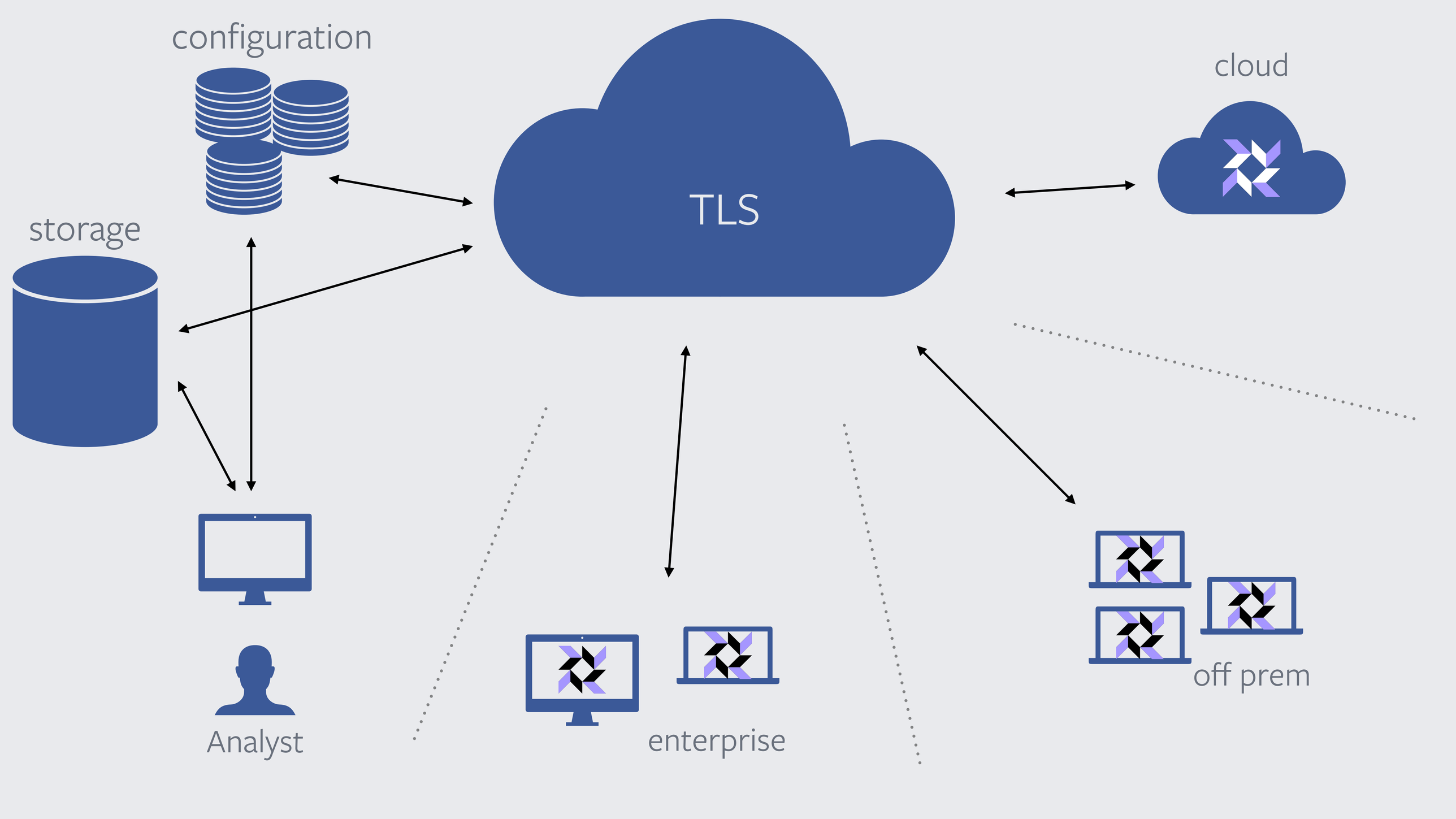
How does osquery work?

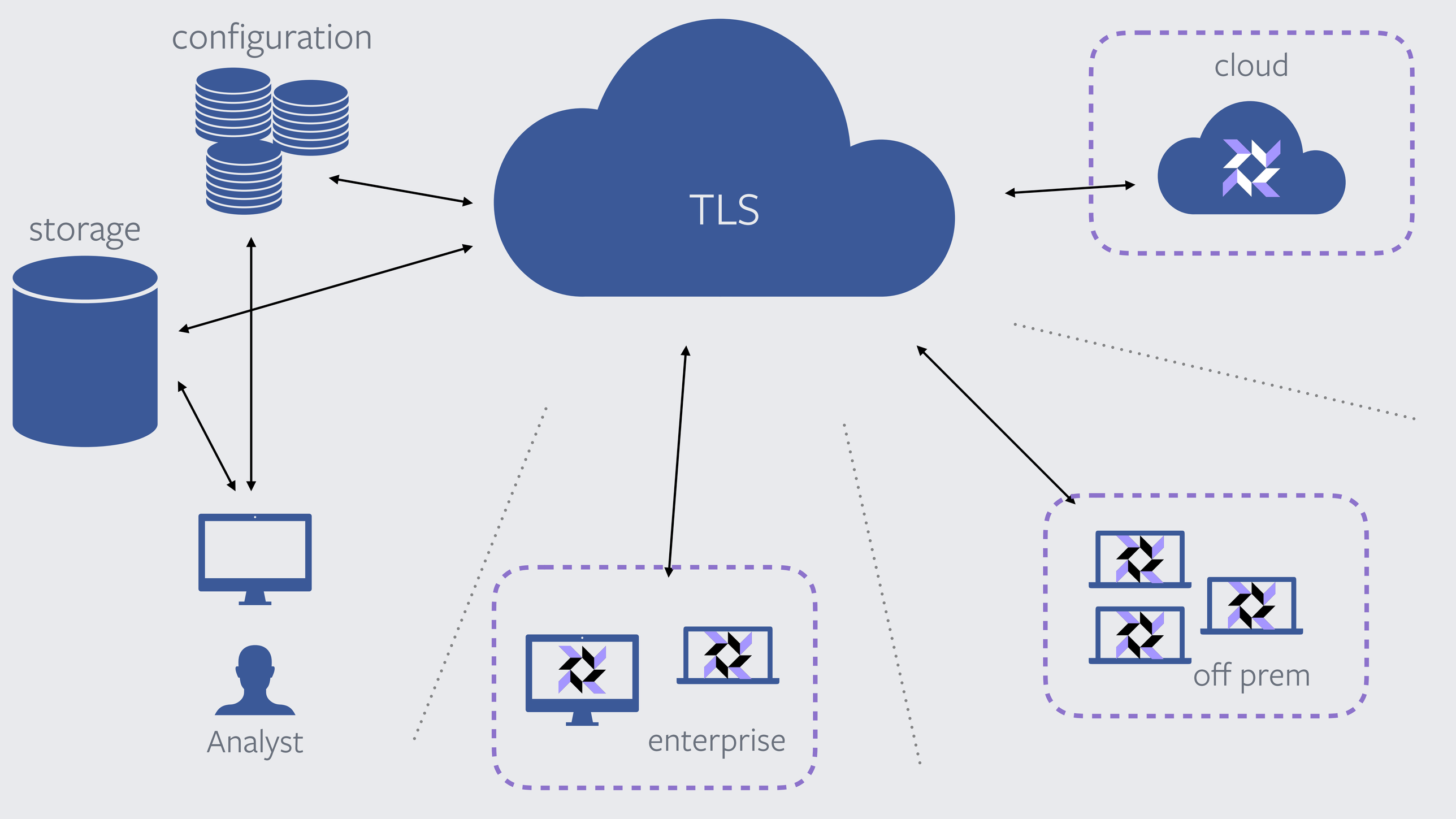


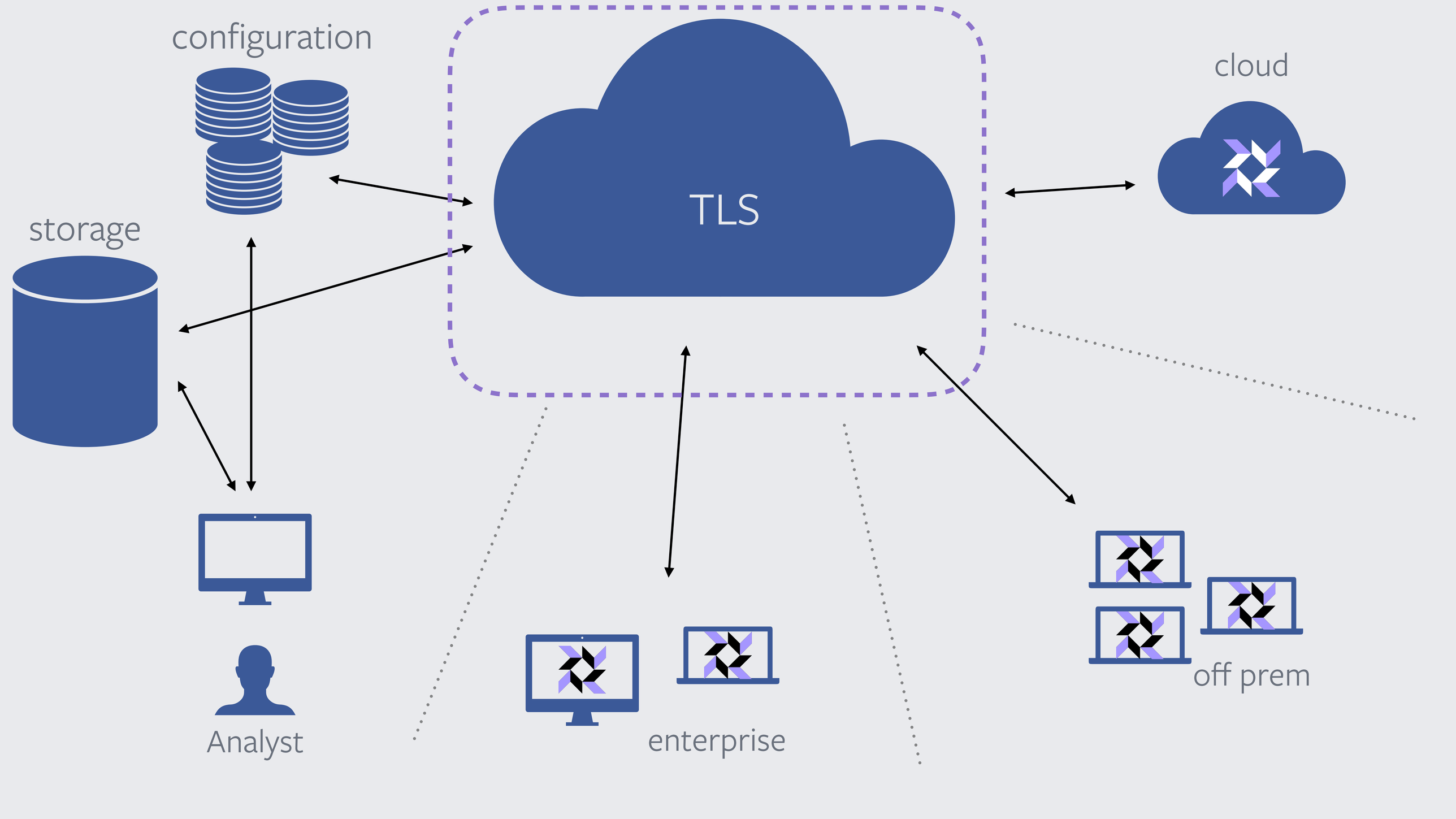
How does osquery work?

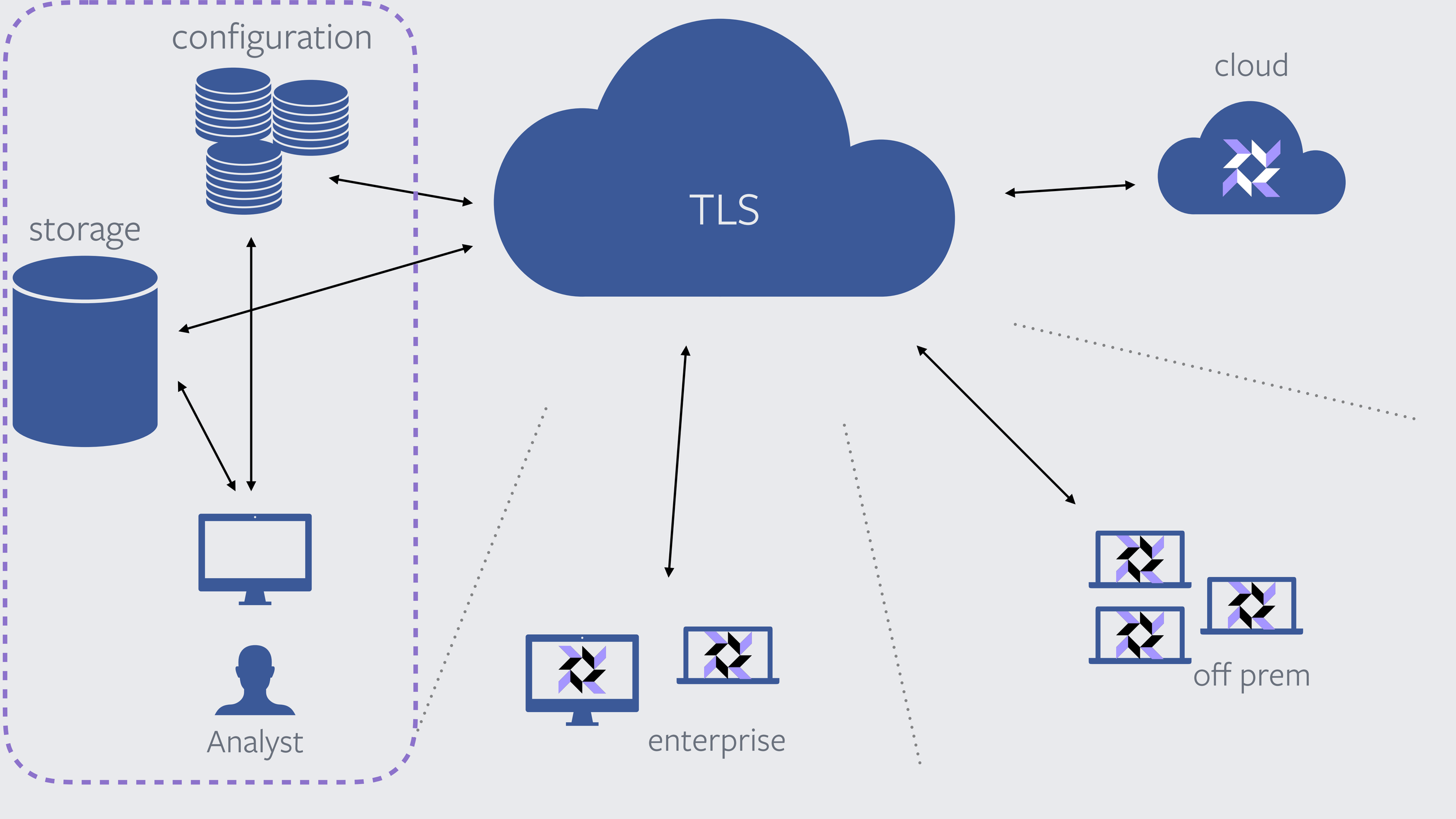


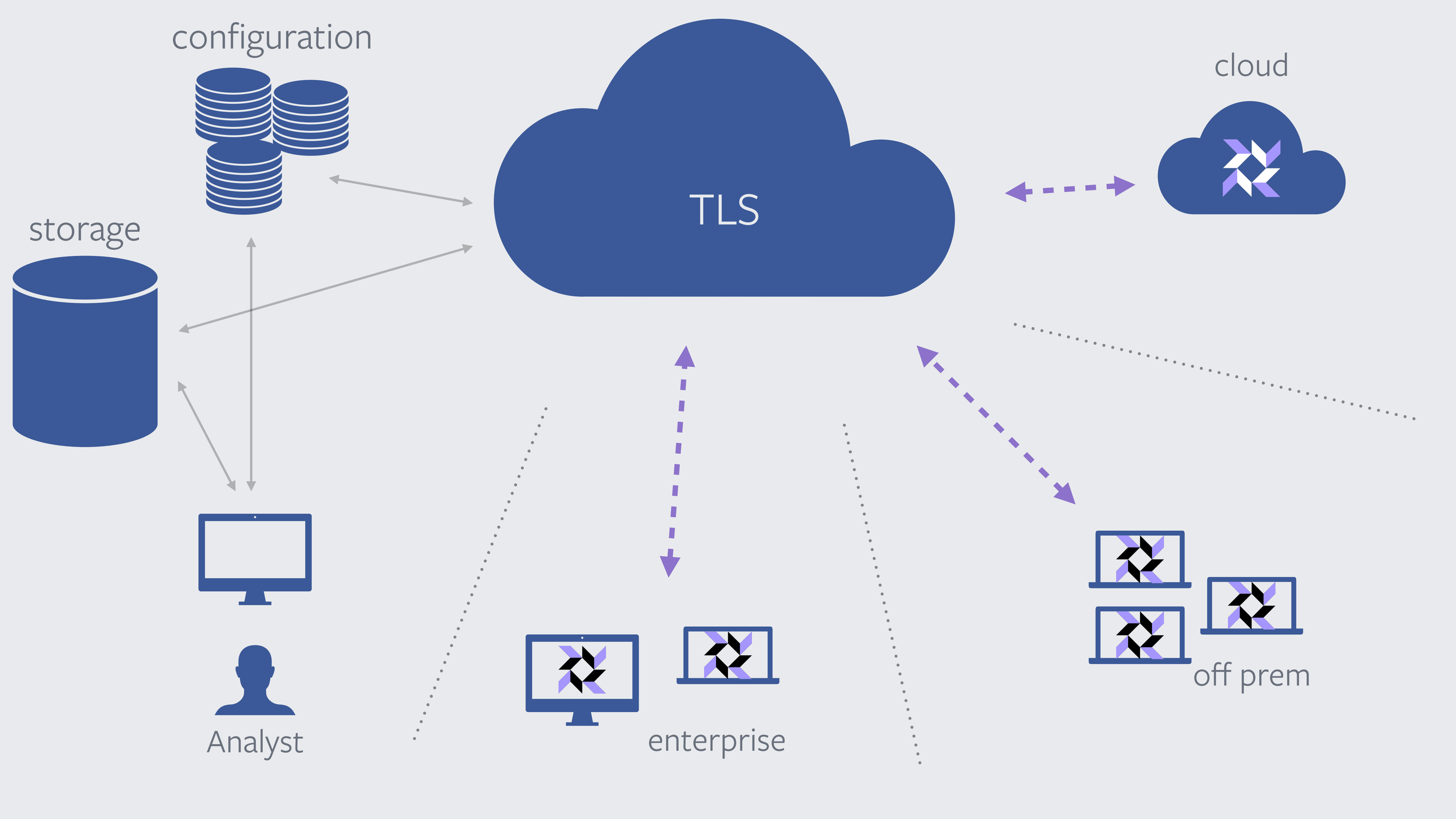
~~ad-hoc~~ ~~distributed~~ ~~live~~
on-demand deployment

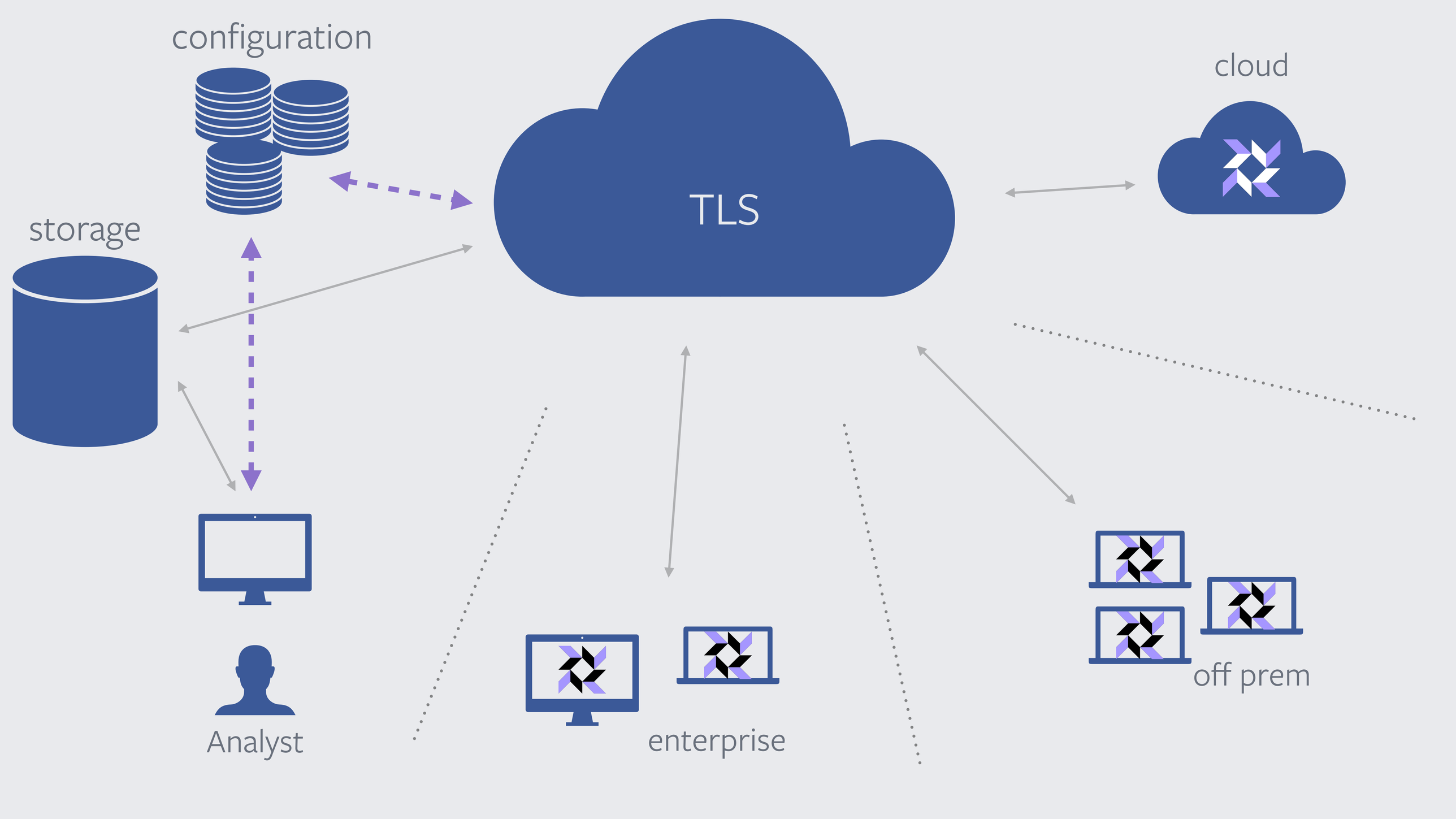

configuration
cloud
TLS
storage
Analyst
enterprise
off prem

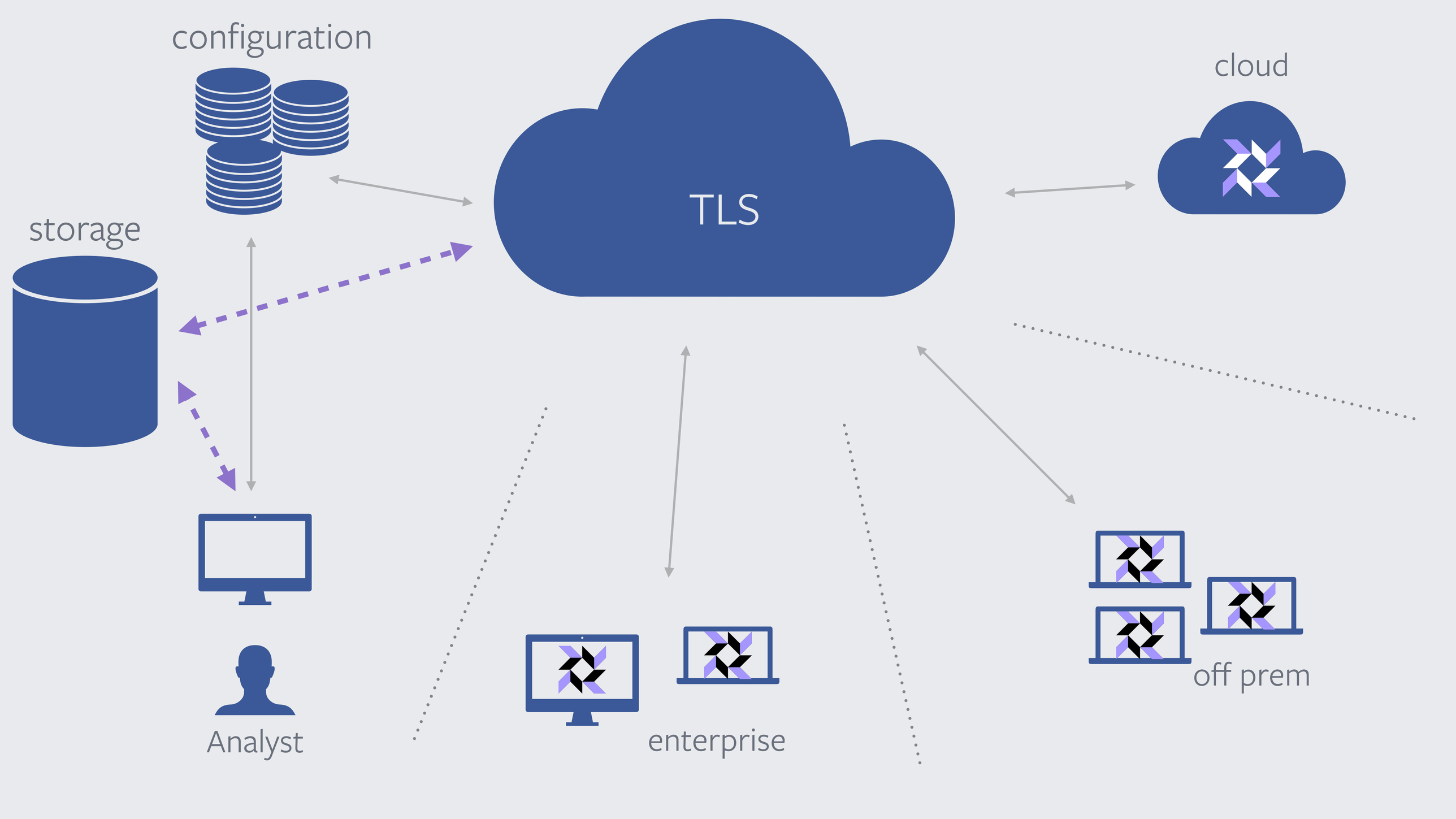

configuration
storage
TLS
cloud
Analyst
enterprise
off prem

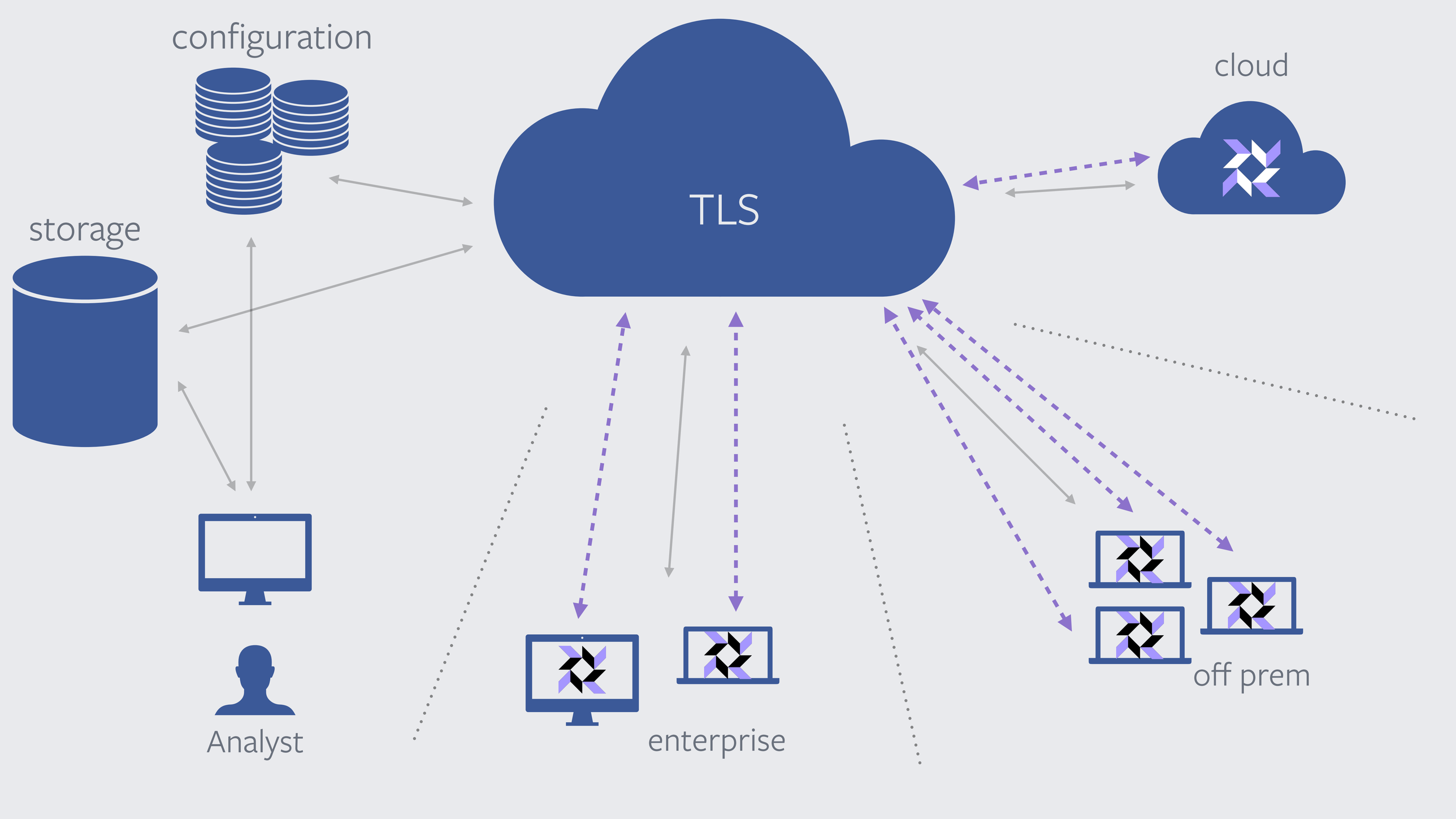

configuration
cloud
TLS
storage
Analyst
enterprise
off prem


configuration
storage
TLS
cloud
Analyst
enterprise
off prem

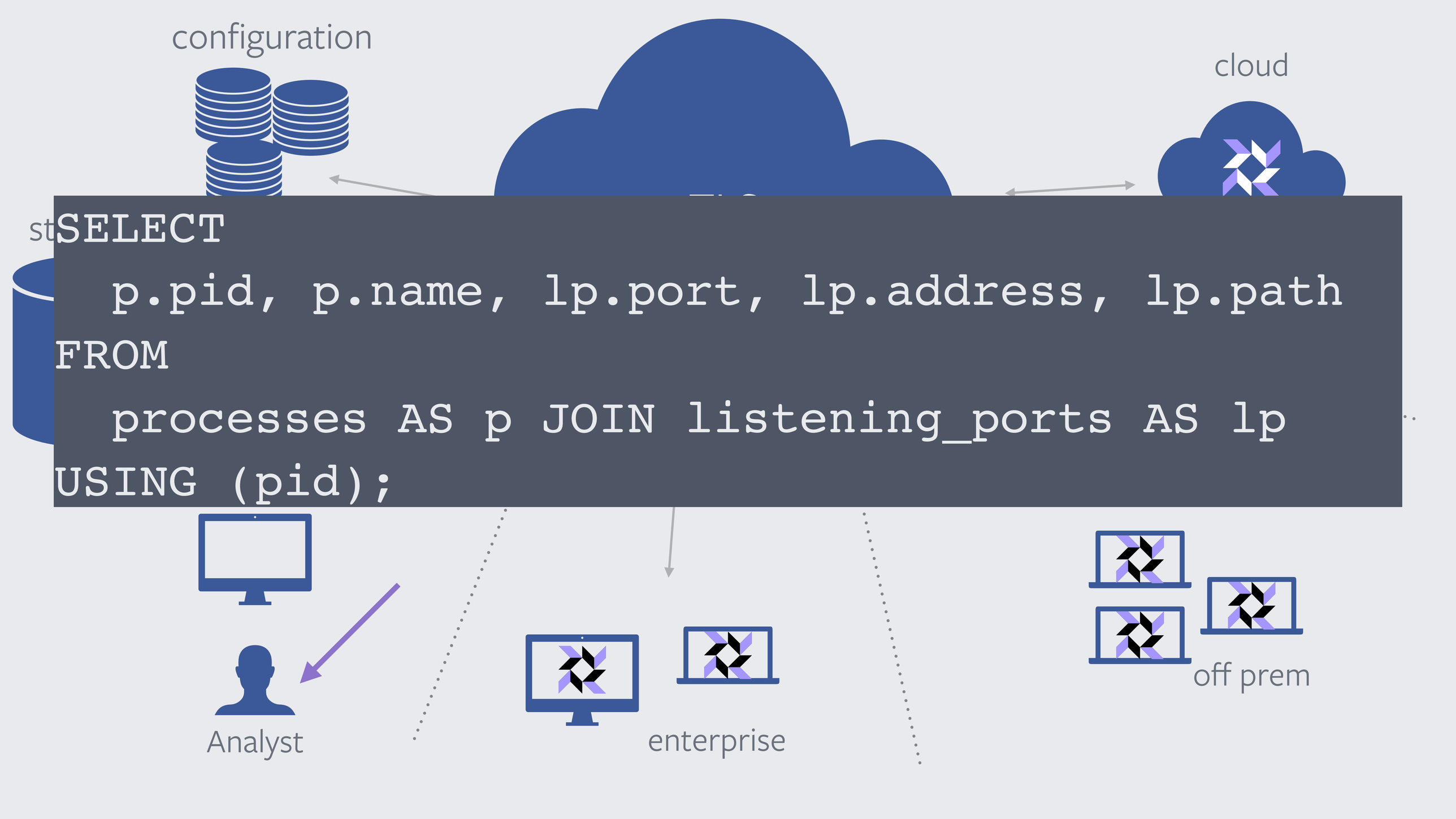

configuration
cloud
TLS
storage
Analyst
enterprise
off prem

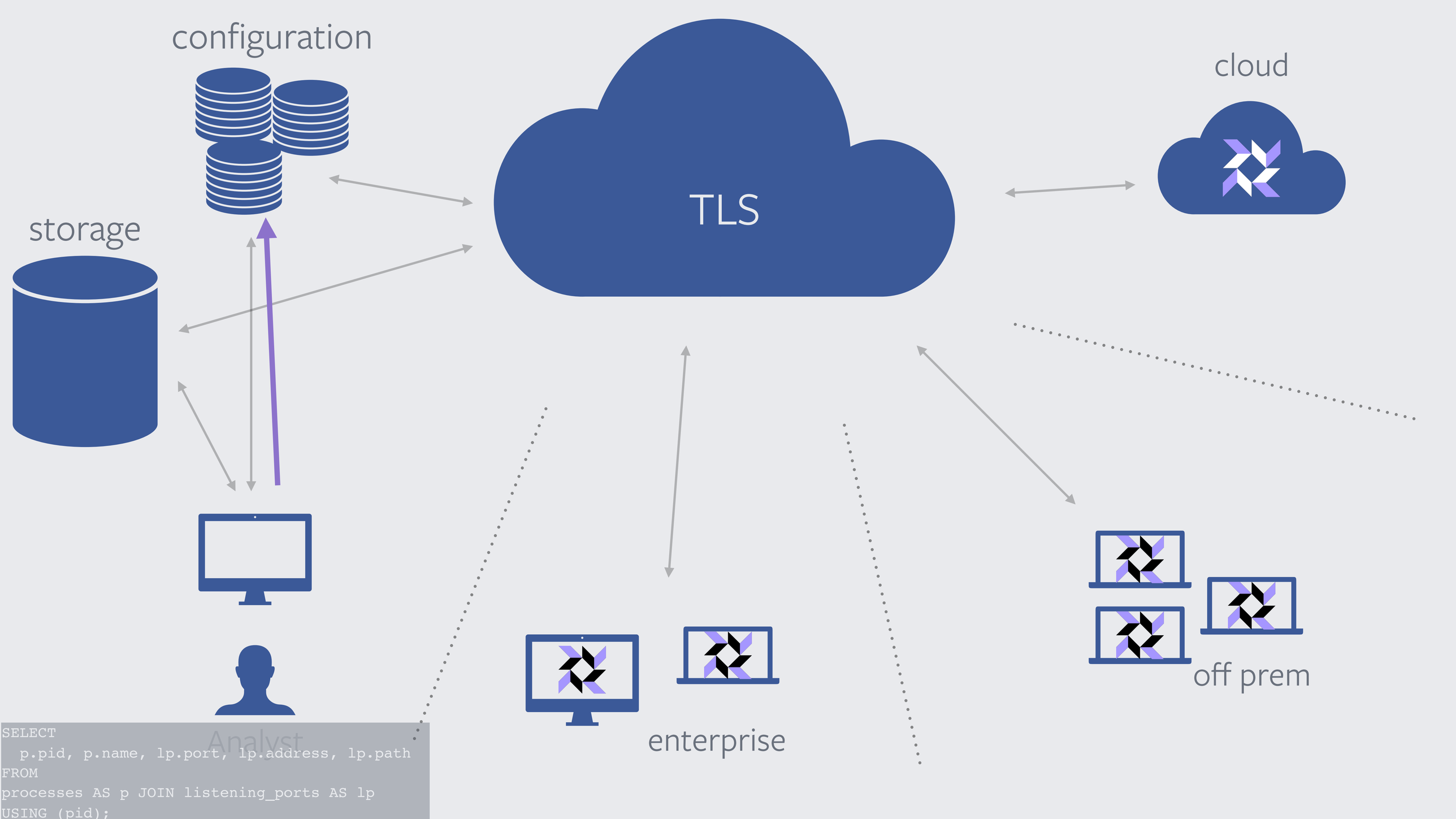

configuration
cloud
TLS
storage
Analyst
enterprise
off prem


configuration
cloud
TLS
storage
Analyst
enterprise
off prem

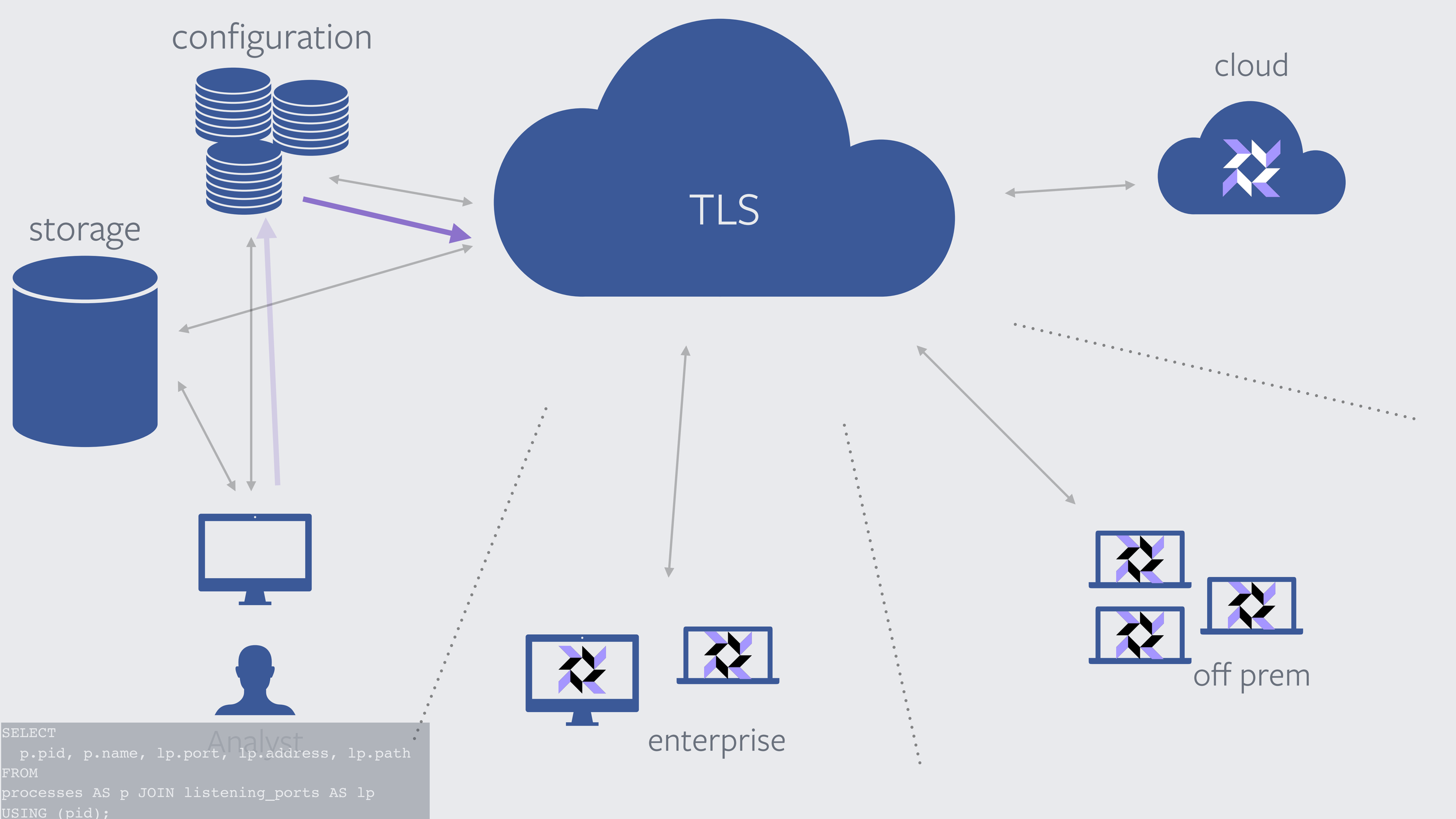

configuration
cloud
TLS
storage
Analyst
enterprise
off prem

distributed queries

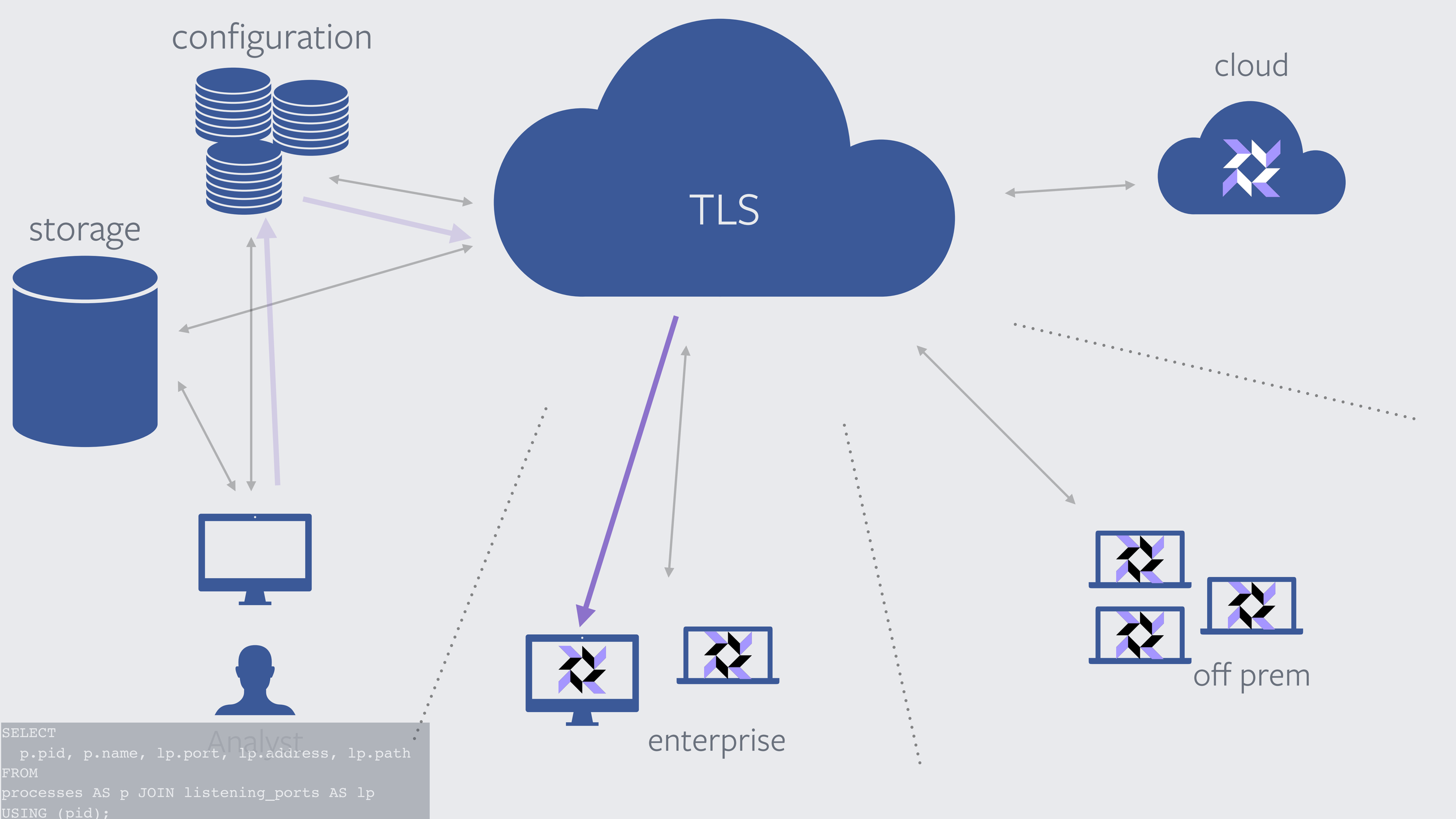

configuration
cloud
st
SELECT
p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
Analyst
enterprise
off prem



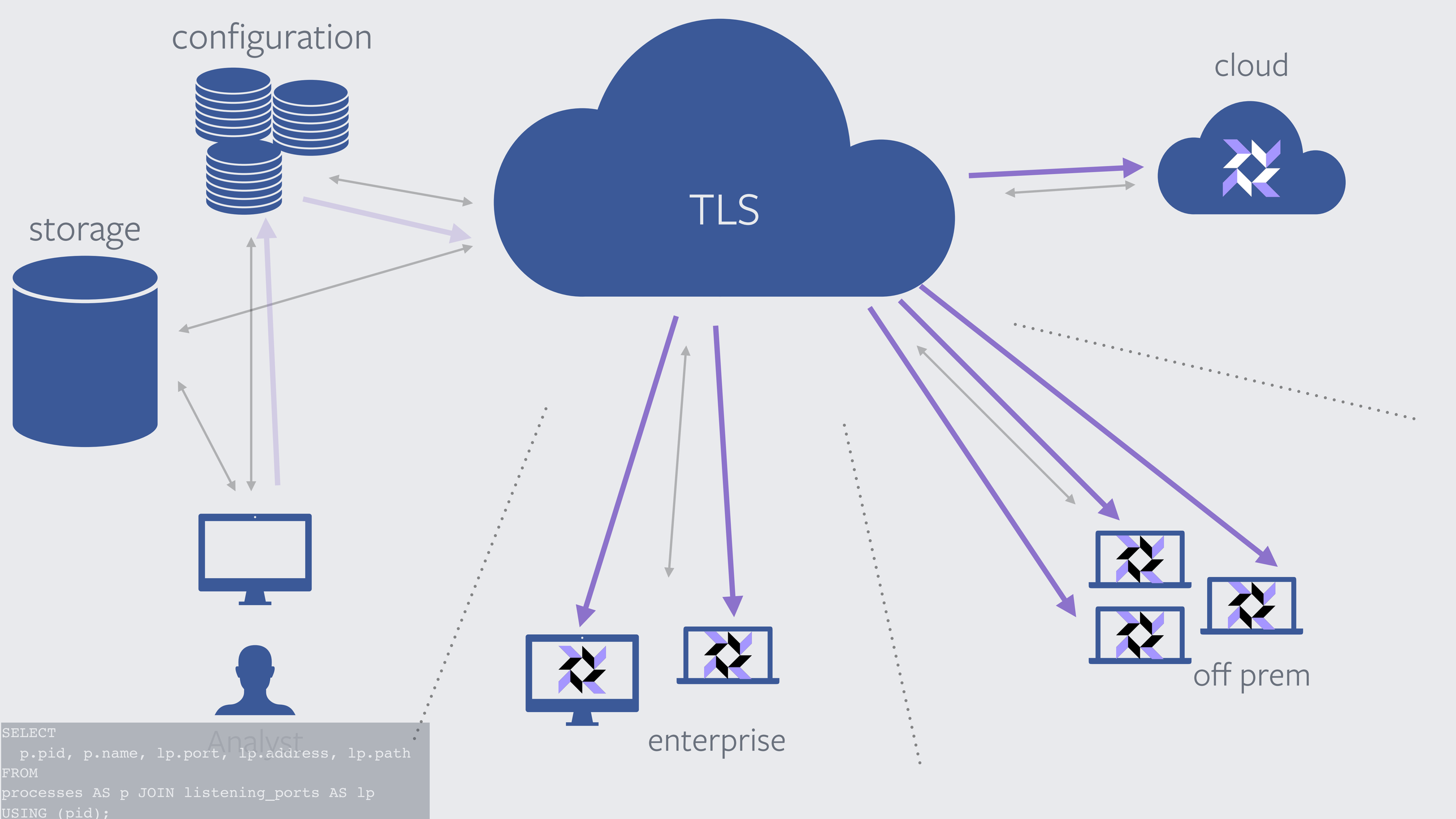

```
SELECT
  p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
```



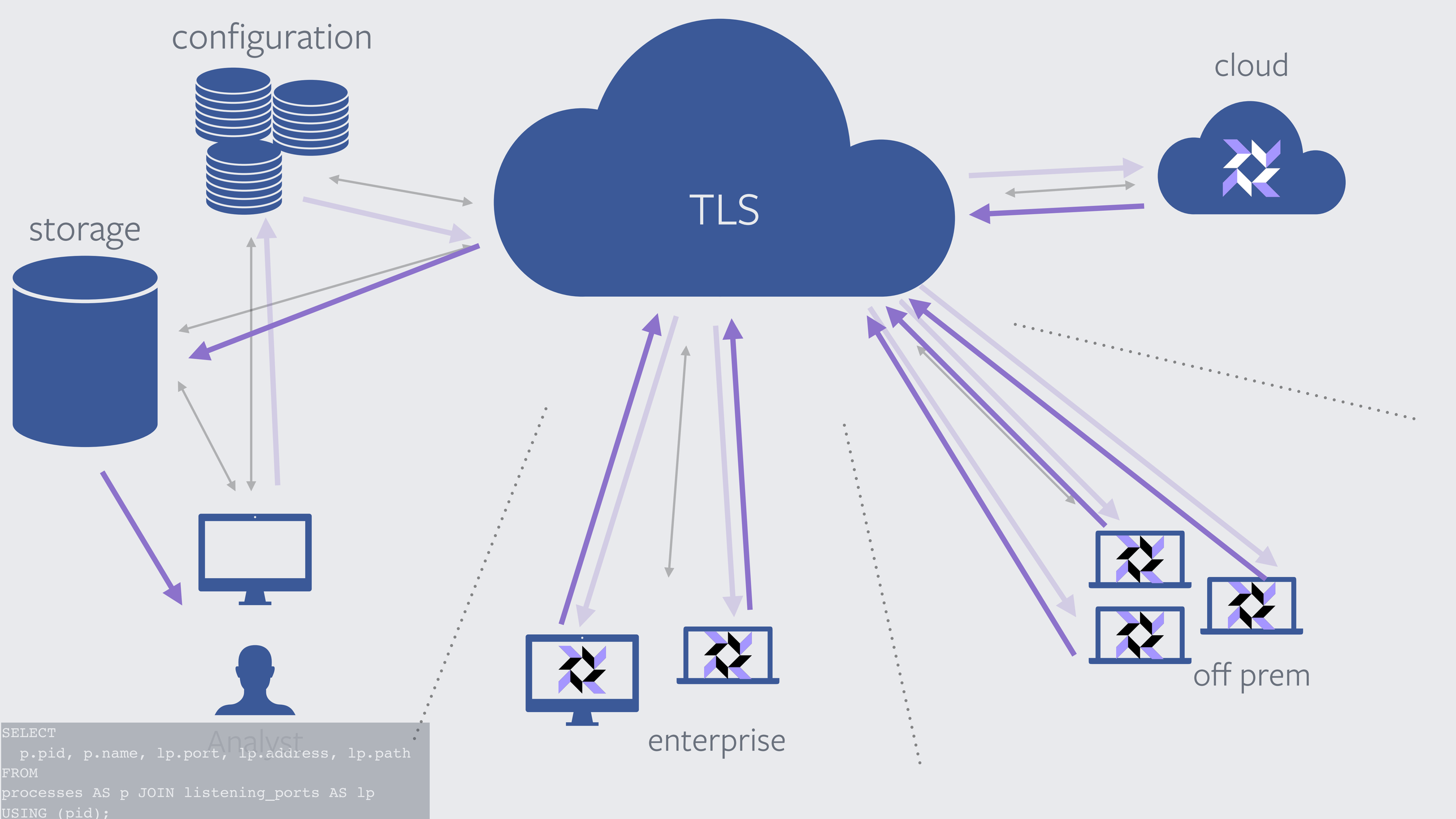

```
SELECT
  p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
```



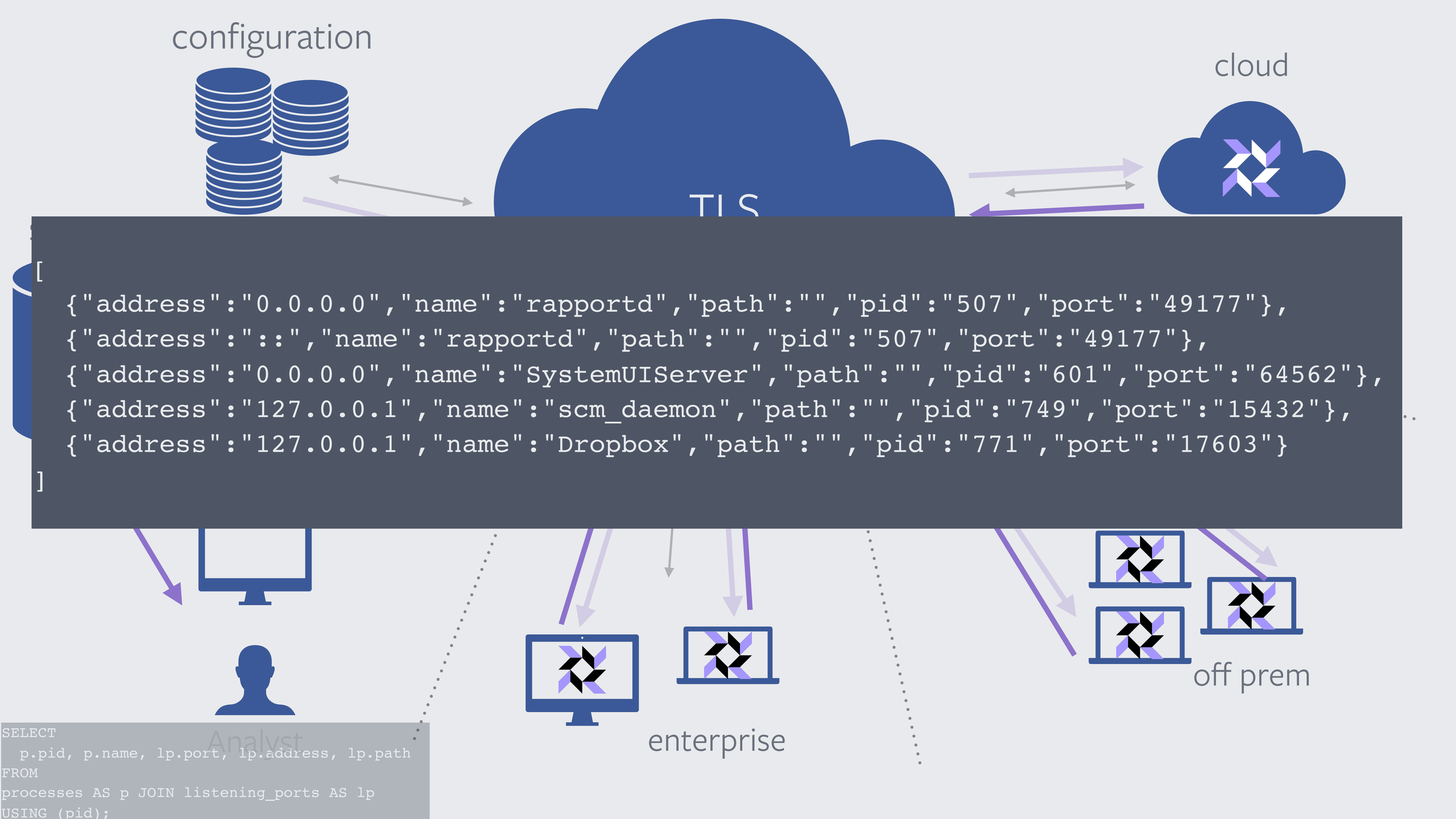

```
SELECT
  p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
```




```
SELECT
  p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
```

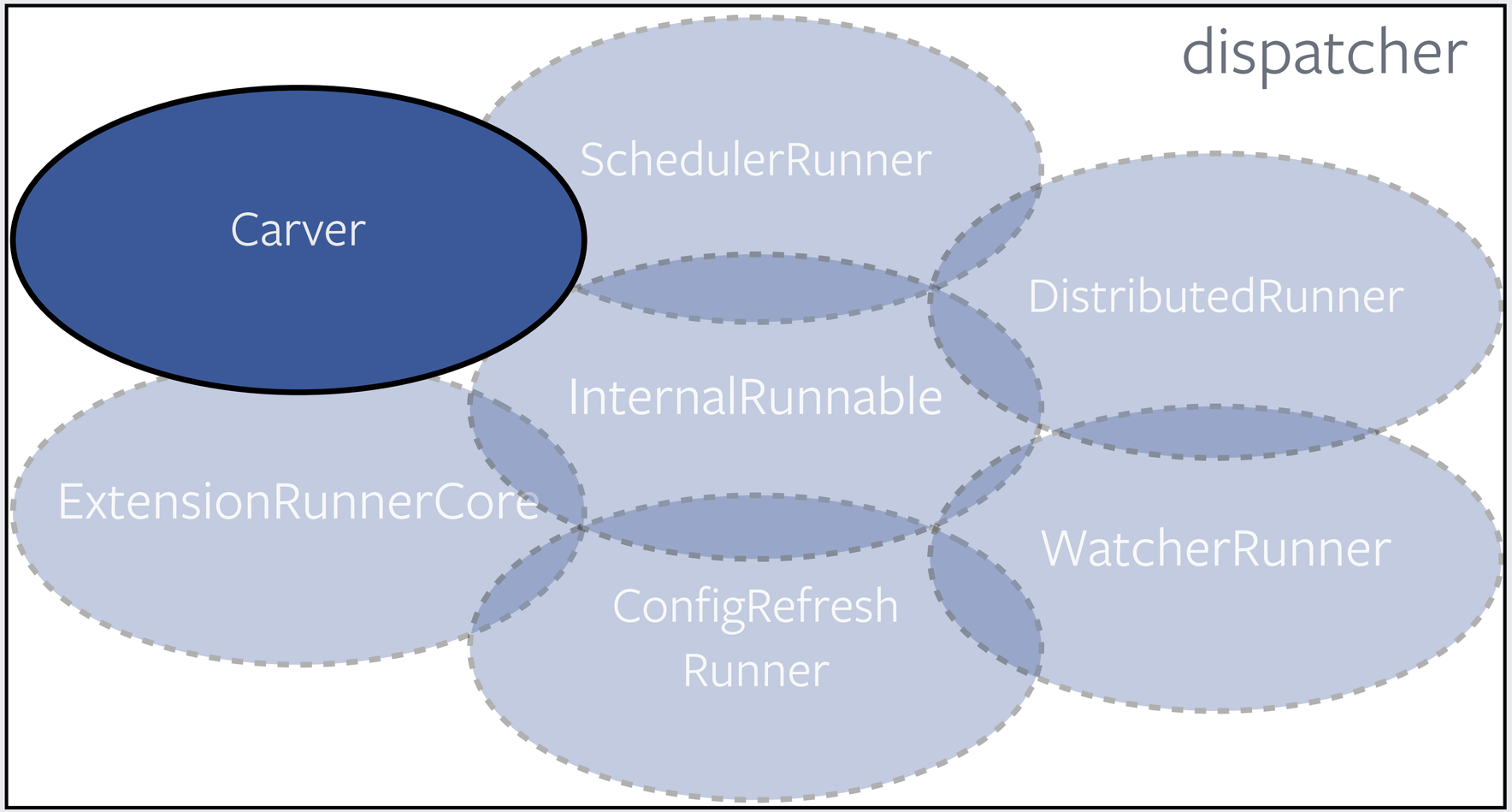



```
SELECT
  p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);
```

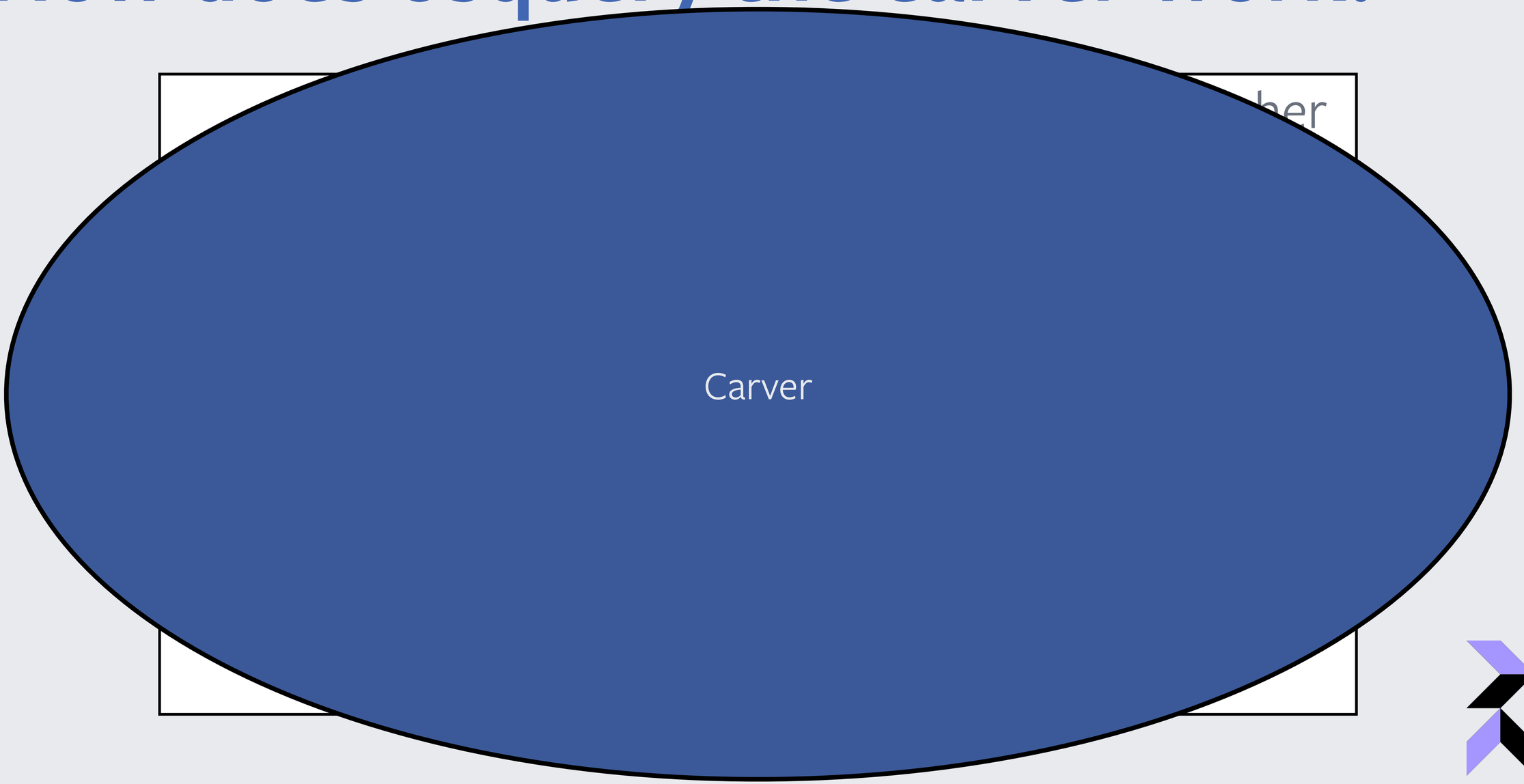

configuration
cloud
TLS
[
{"address":"0.0.0.0","name":"rapportd","path":"","pid":"507","port":"49177"},
{"address":"::","name":"rapportd","path":"","pid":"507","port":"49177"},
{"address":"0.0.0.0","name":"SystemUIServer","path":"","pid":"601","port":"64562"},
{"address":"127.0.0.1","name":"scm_daemon","path":"","pid":"749","port":"15432"},
{"address":"127.0.0.1","name":"Dropbox","path":"","pid":"771","port":"17603"}
]
Analyst
enterprise
off prem
SELECT
p.pid, p.name, lp.port, lp.address, lp.path
FROM
processes AS p JOIN listening_ports AS lp
USING (pid);

Repurposing Runnables

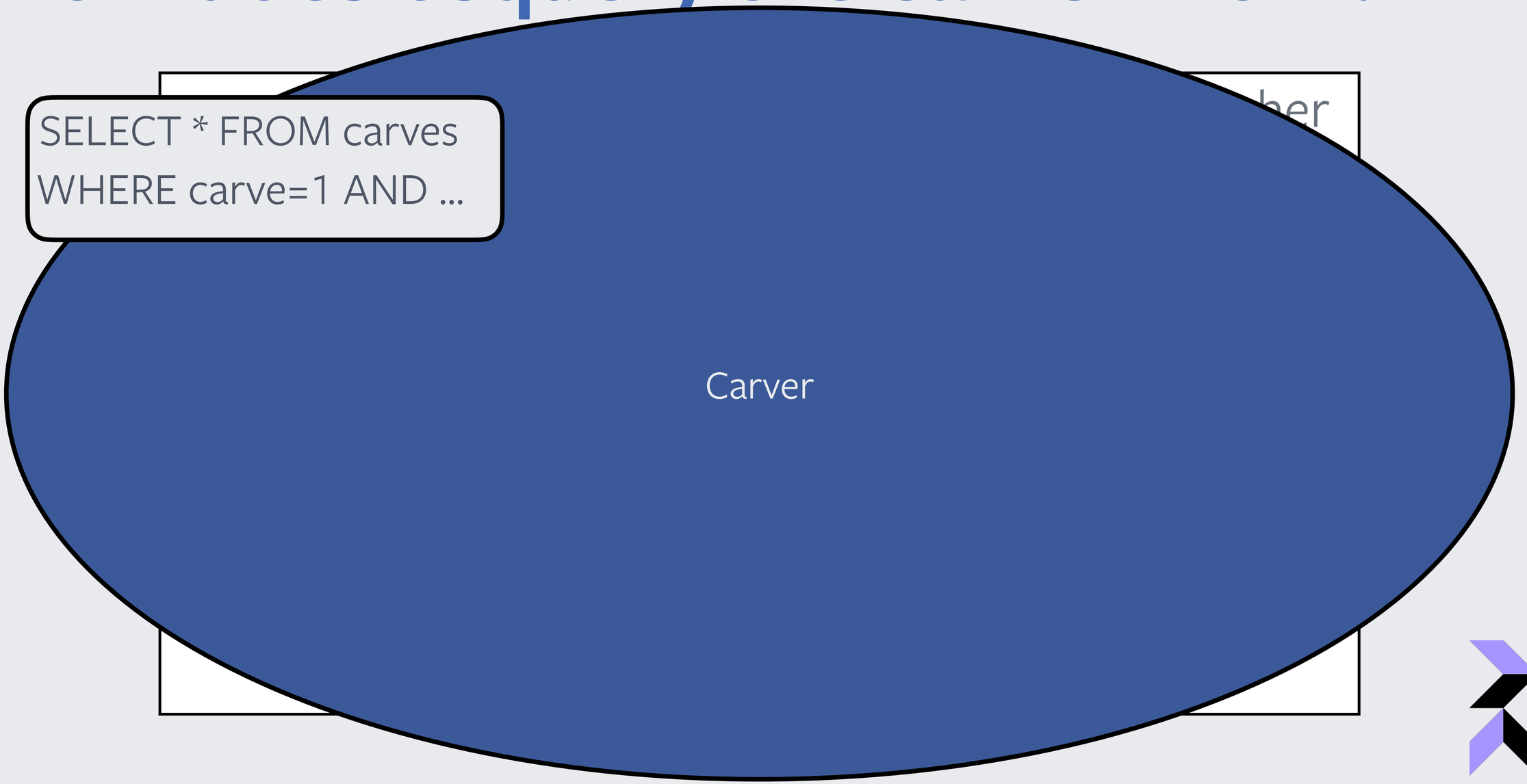
How does osquery work?



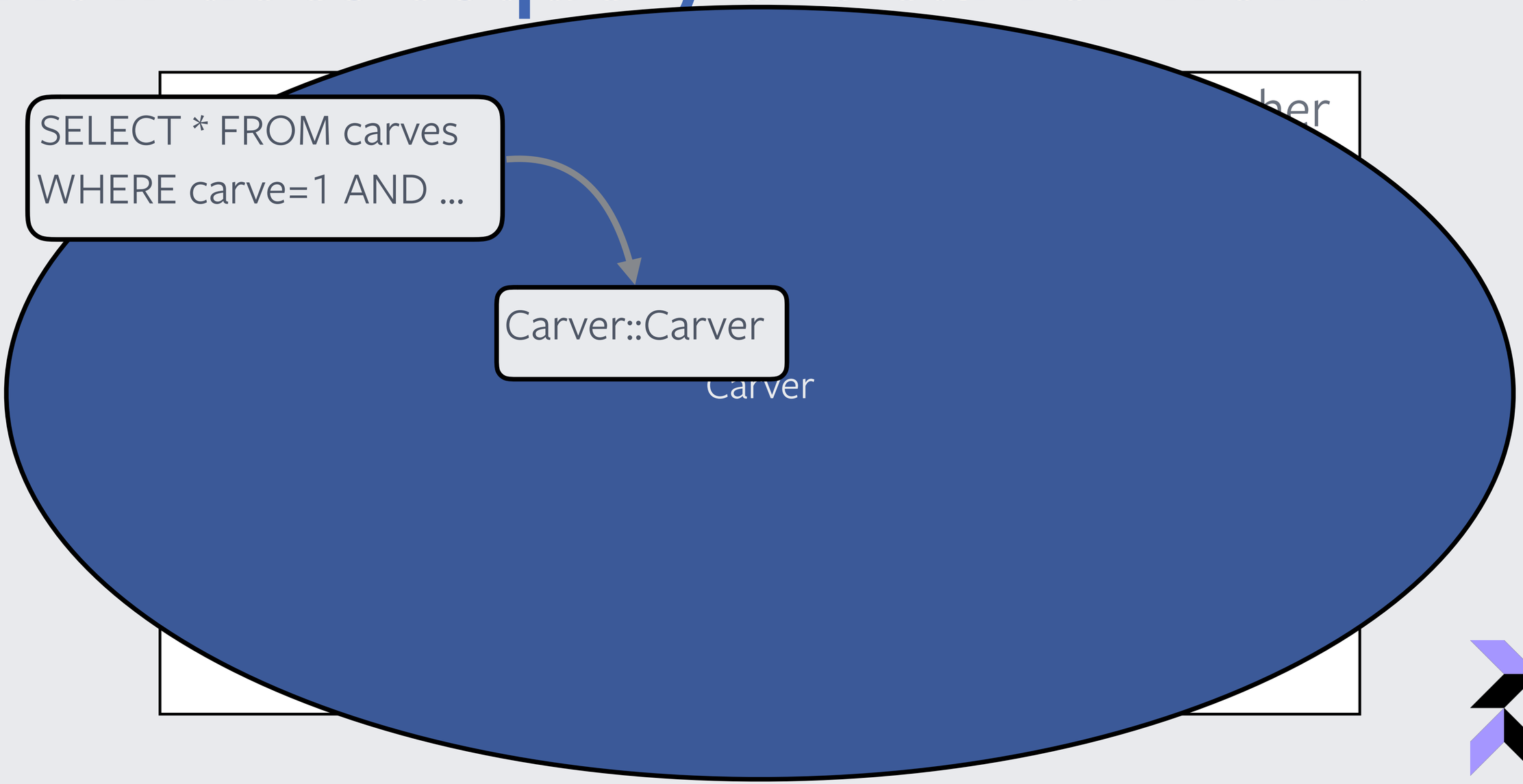
How does ~~osquery~~ the carver work?



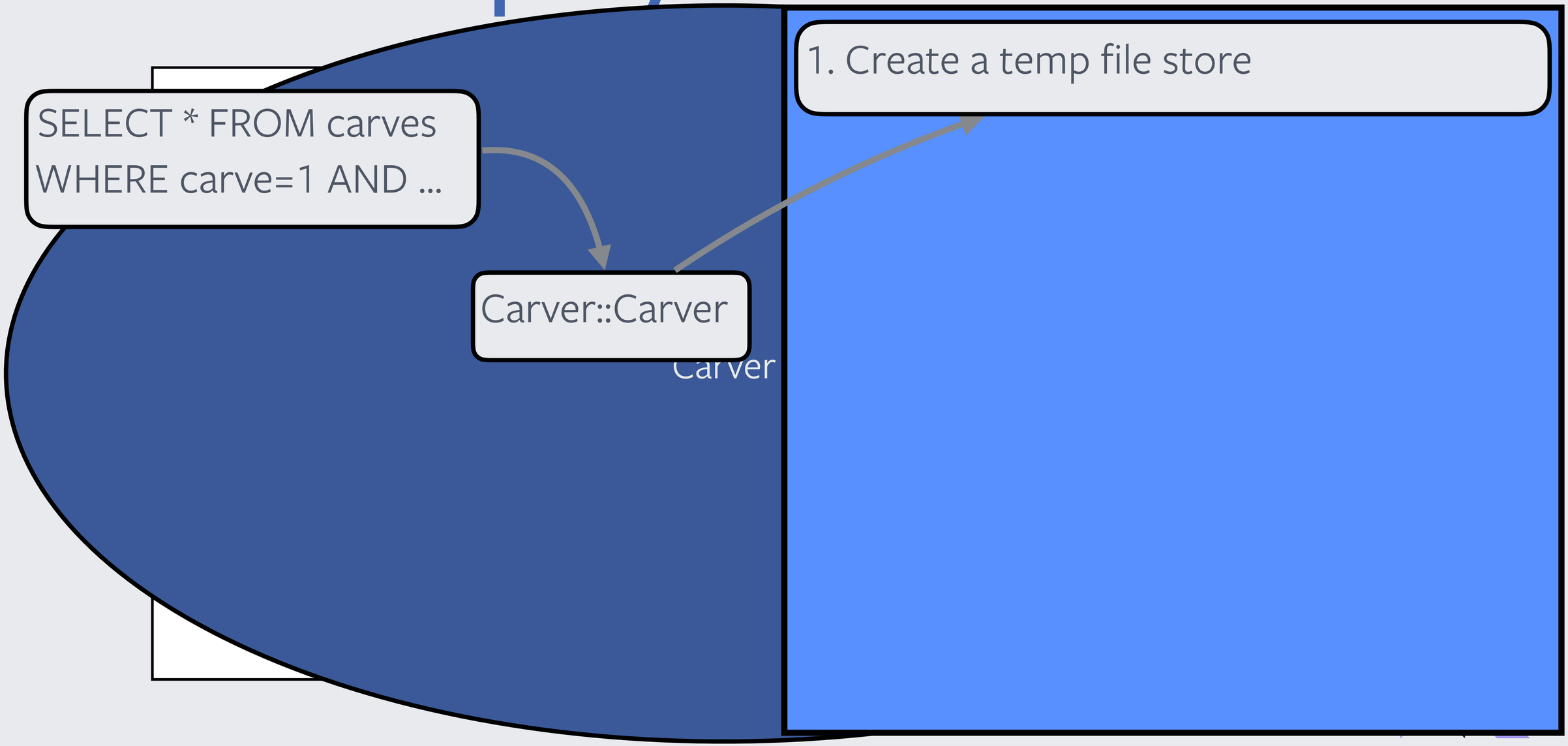
How does ~~osquery~~ the carver work?



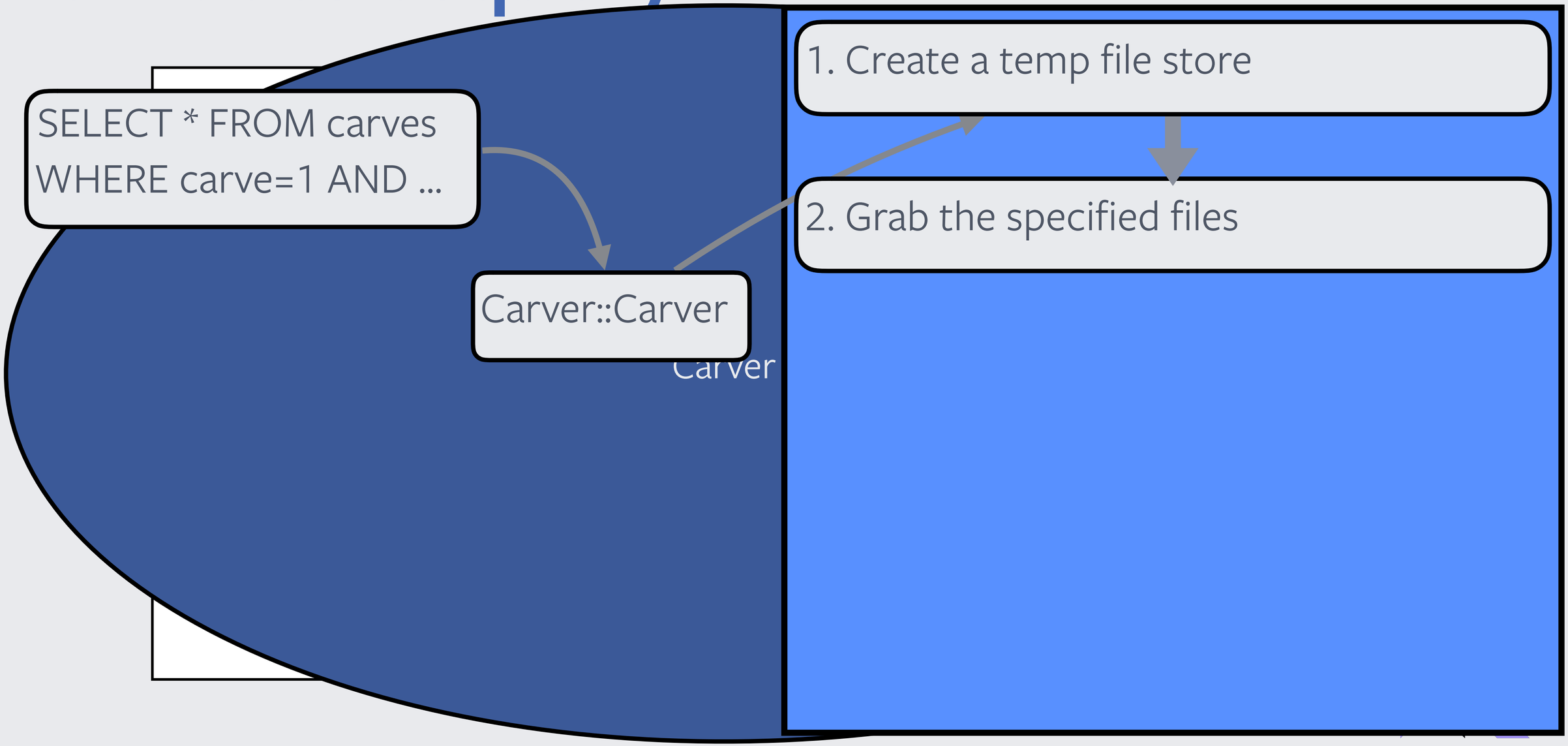
How does ~~osquery~~ the carver work?



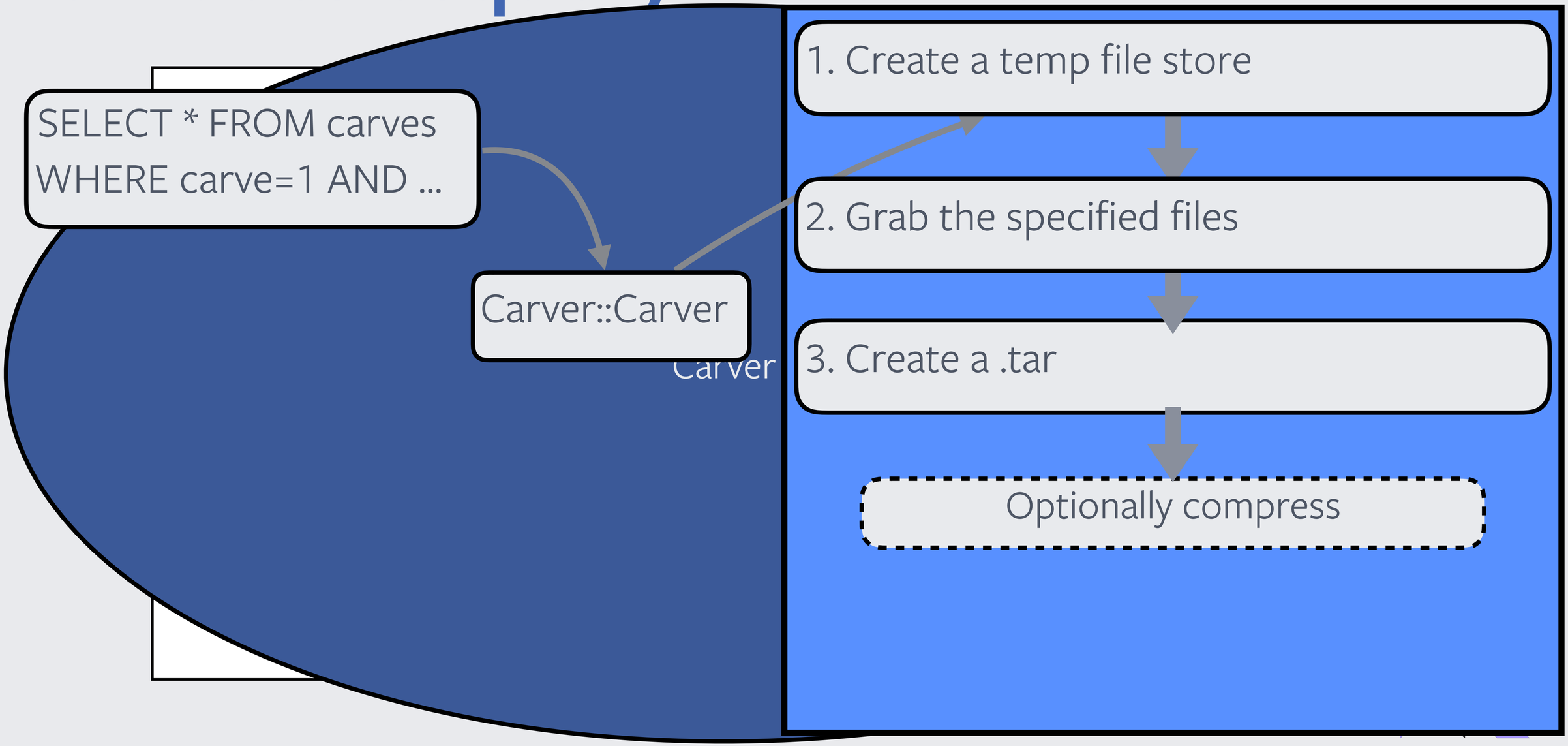
How does ~~osquery~~ the carver work?



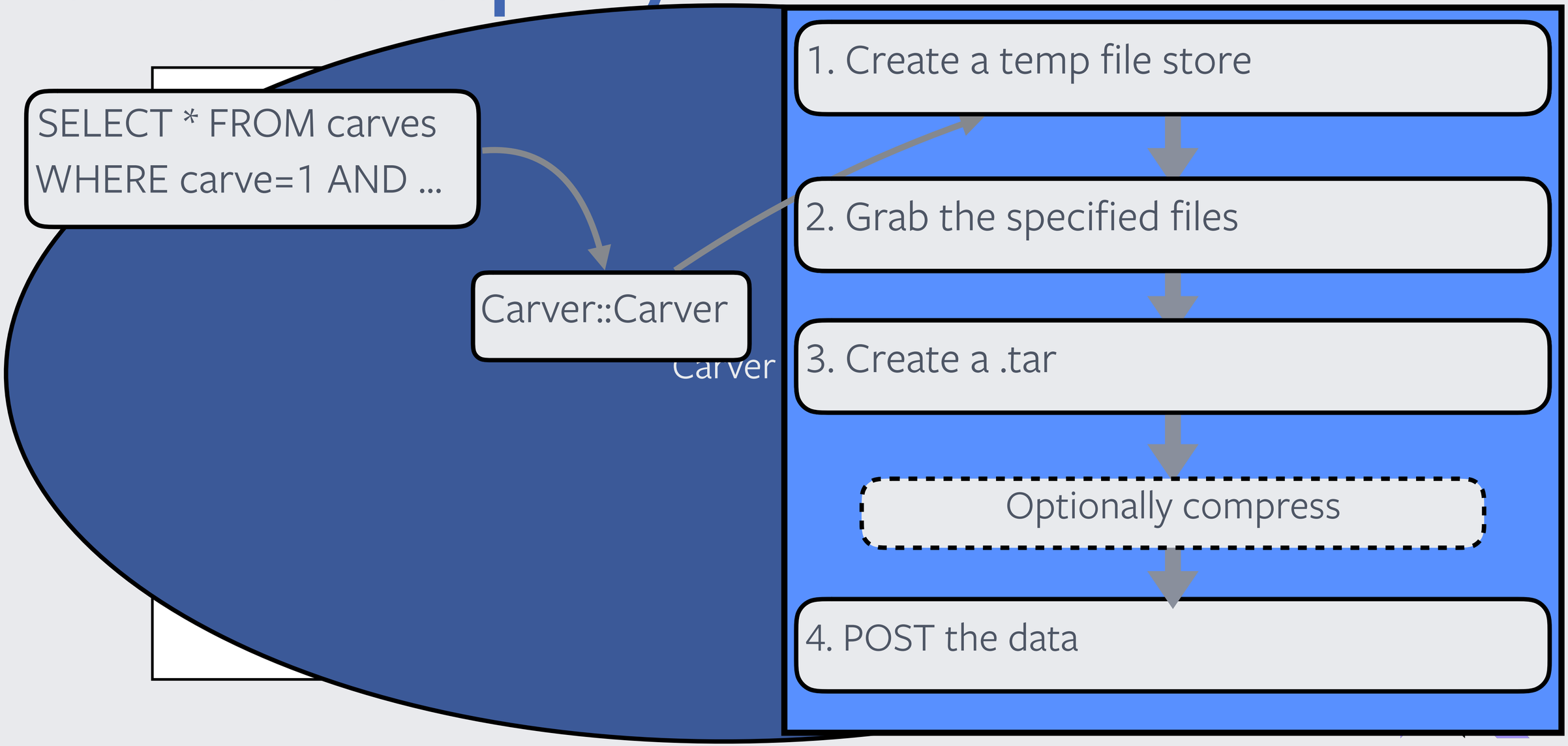
How does ~~osquery~~ the carver work?



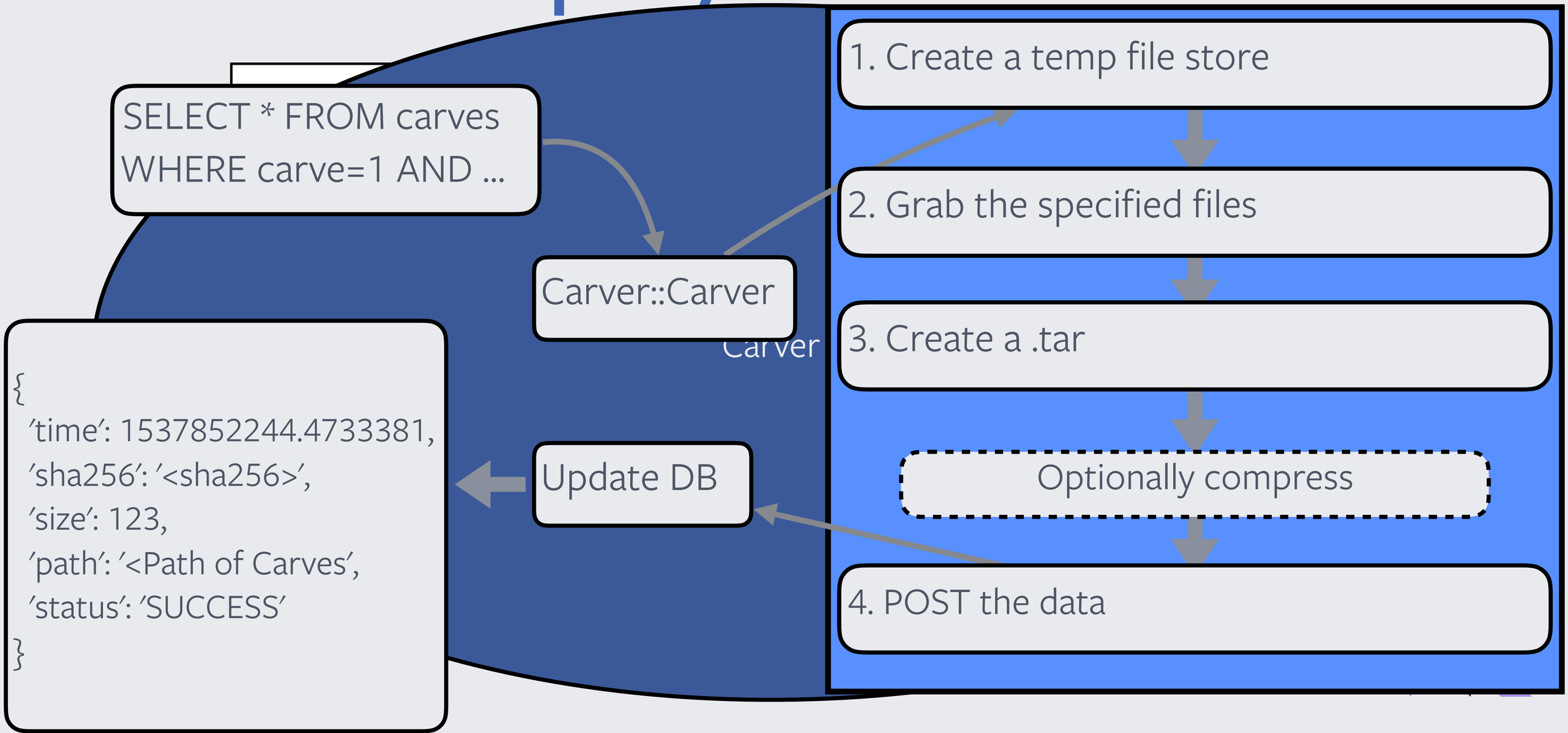
How does ~~osquery~~ the carver work?



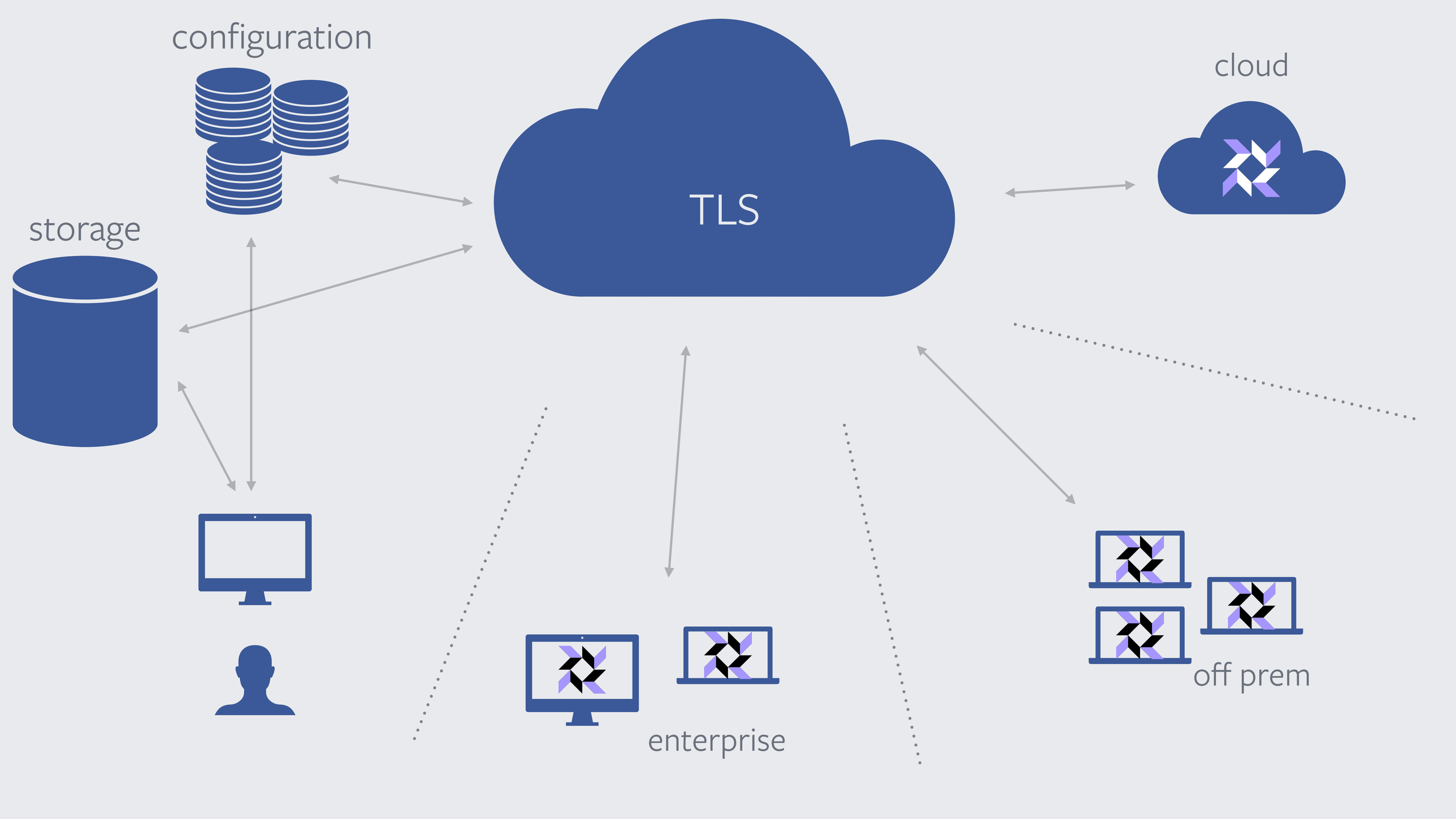
How does ~~osquery~~ the carver work?

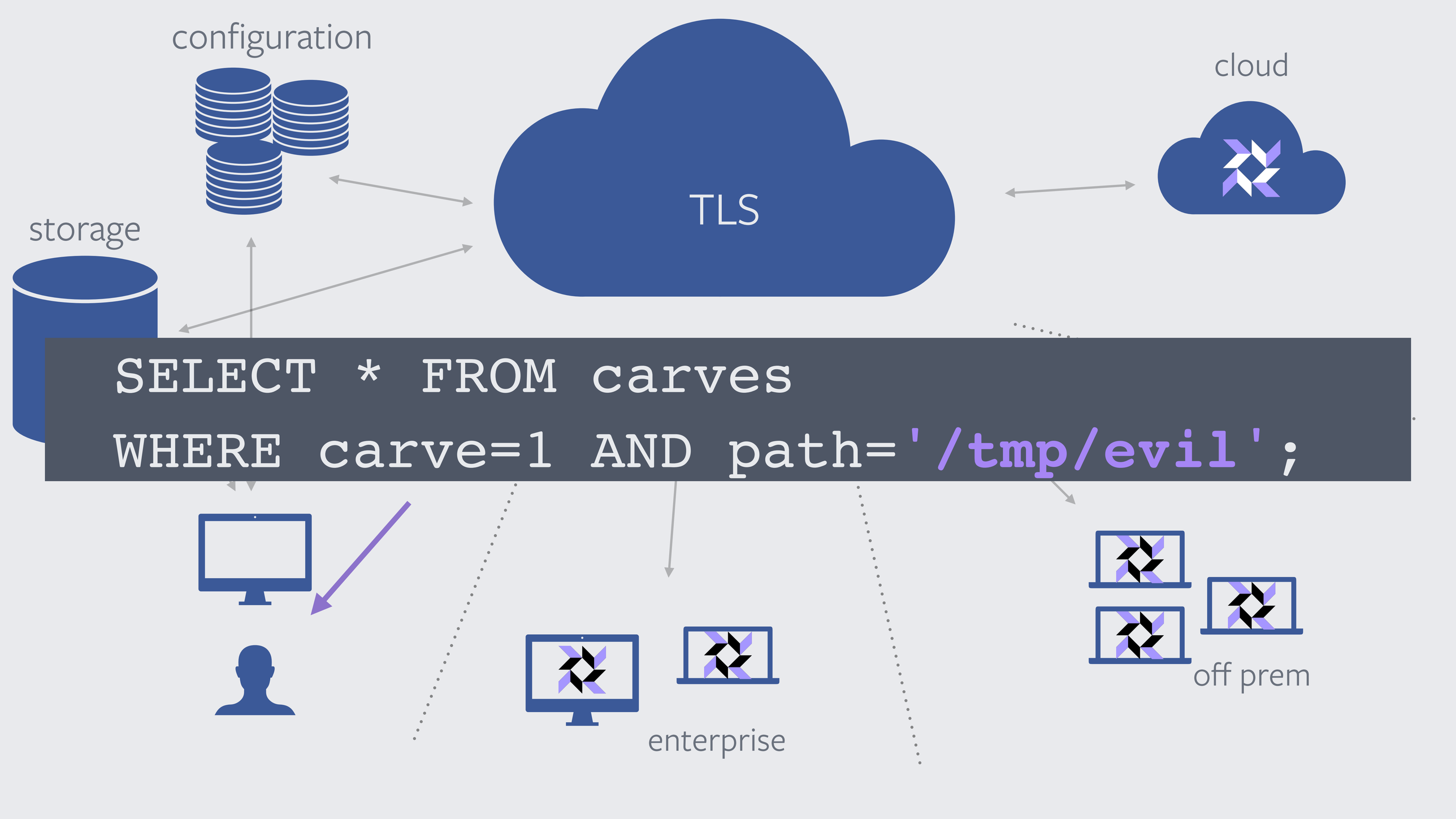


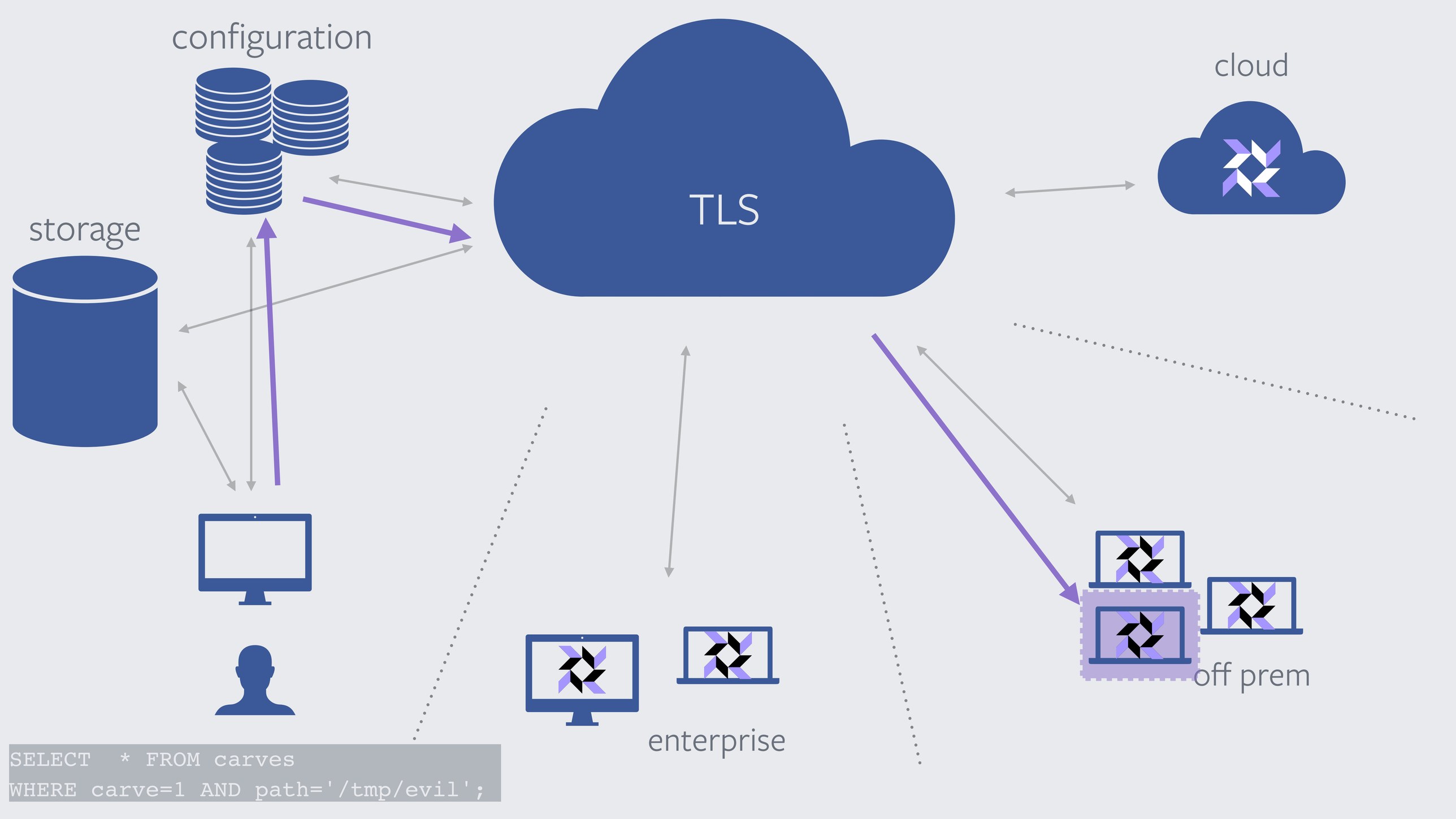
How does ~~osquery~~ the carver work?



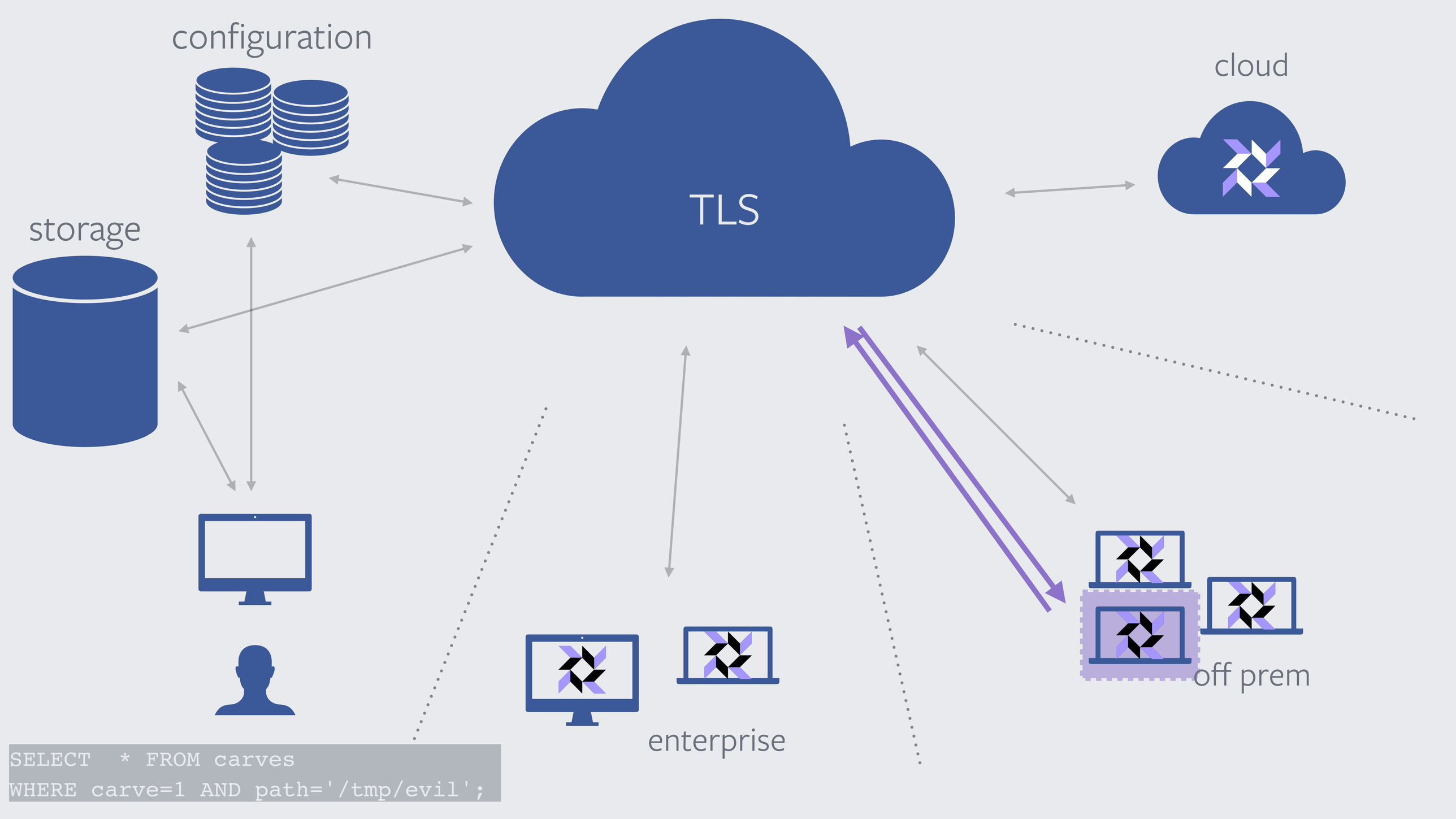
The file carver

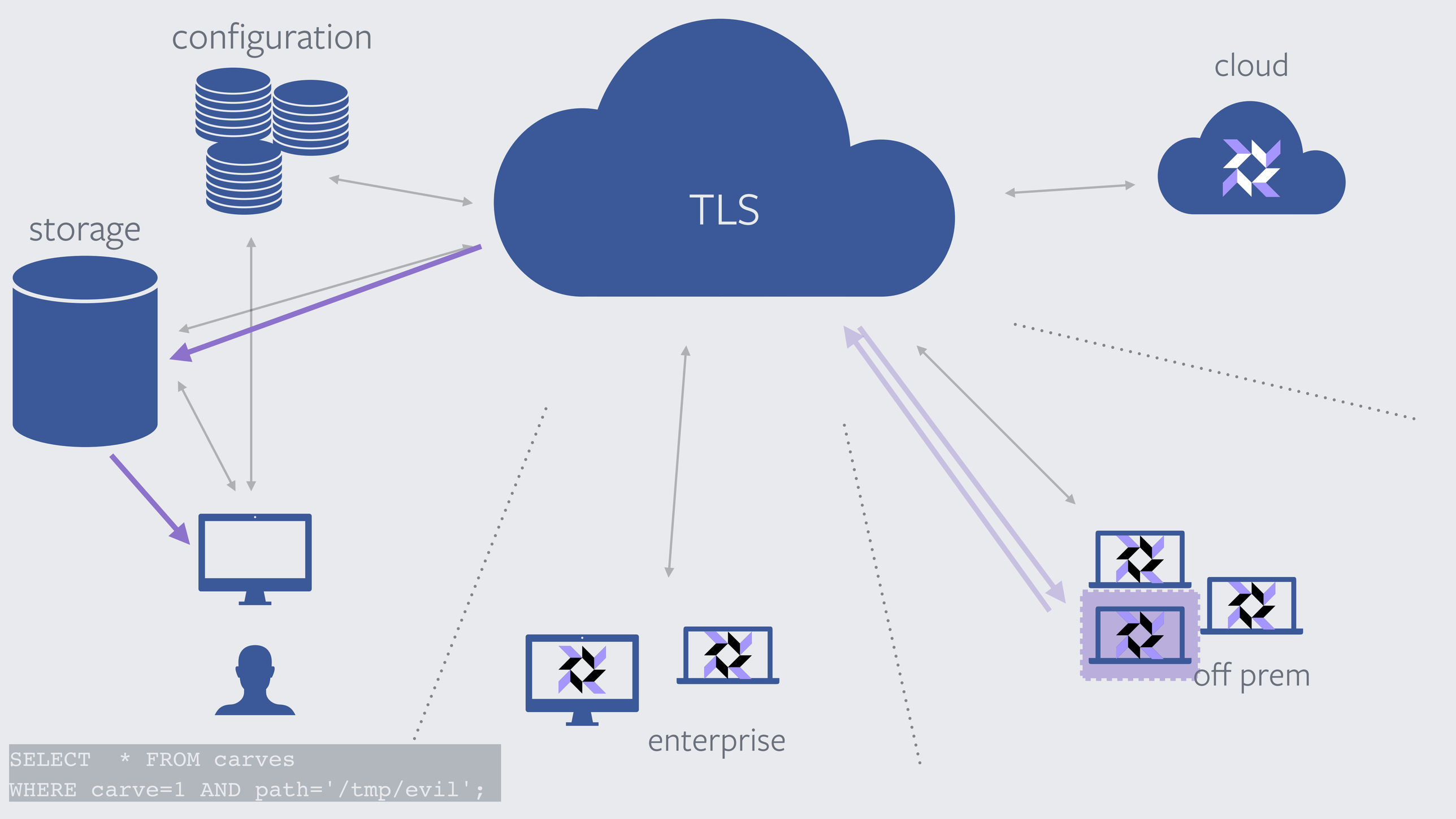

configuration
cloud
TLS
storage
enterprise
off prem


configuration
cloud
TLS
storage
SELECT * FROM carves
WHERE carve=1 AND path='/tmp/evil';
enterprise
off prem




```
SELECT  * FROM carves
WHERE carve=1 AND path='/tmp/evil';
```


configuration
cloud
TLS
storage
enterprise
off prem
SELECT * FROM carves
WHERE carve=1 AND path='/tmp/evil';


configuration
cloud
TLS
storage
enterprise
off prem
SELECT * FROM carves
WHERE carve=1 AND path='/tmp/evil';

Use Cases

Use Cases

- Auto Carving
  - New files
  - Case artifacts

Use Cases

- Auto Carving
  - New files
  - Case artifacts
- Asynchronous acquisitions
  - Endpoint is off corp network
  - Laptop is shut and then opened

Surprises

Carver Wins

```
SELECT
  *
FROM
  carves
WHERE
  carve=1 AND
path LIKE '/Users/%/Downloads/%';
```

Limitations

Carver Limitations

- Watchdog and size limits

Carver Limitations

- Watchdog and size limits
- ″carves″

Carver Limitations

- Watchdog and size limits
- ″carves″
- Block POST retry

Carver Limitations

- Watchdog and size limits
- ″carves″
- Block POST retry
- Limited to TLS endpoints

Carver Limitations

- Watchdog and size limits
- ″carves″
- Block POST retry
- Limited to TLS endpoints
- How does one carve?

Carver Endpoints

Carver Endpoints

- Zentral
  - github.com/zentralopensource/zentral
- SGT
  - github.com/OktaSecurityLabs/sgt

Carver ~~Endpoints~~ Specs and Docs

- Doorman issue
  - github.com/mwielgoszewski/doorman/issues/120
- osquery integration test
  - osquery/tools/tests/test_http_server.py

Thanks! Questions?

- Nick Anderson
  - Security Engineer at Facebook
- thor@fb.com
  - Super legit, not an alias
- Github - github.com/muffins
- Twitter - twitter.com/poppyseedplehzr
- Slack - thor

facebook