# A tale of migrating to Puppet 3.8 and Multi Datacenter Puppet Infrastructure

Alexandros Kosiaris <akosiaris@wikimedia.org>

GRNOG - 4th Technical Meeting

02 Dec 2016

# Wikimedia Foundation

What is the WMF?

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors
- 280 employees ( 60-70 SWE, 20 ops)

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors
- 280 employees ( 60-70 SWE, 20 ops)
- Number 5 in Top 10 website companies

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors
- 280 employees ( 60-70 SWE, 20 ops)
- Number 5 in Top 10 website companies
- 4 Datacenters. 2 main ones, 2 caching, avg latency  50ms

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors
- 280 employees ( 60-70 SWE, 20 ops)
- Number 5 in Top 10 website companies
- 4 Datacenters. 2 main ones, 2 caching, avg latency  50ms
- Server count: ~1200

# Wikimedia Foundation

What is the WMF?

- Non-profit organization focusing on free/open content
- No ads, no VC money
- Entirely funded by small donors
- 280 employees ( 60-70 SWE, 20 ops)
- Number 5 in Top 10 website companies
- 4 Datacenters. 2 main ones, 2 caching, avg latency  50ms
- Server count: ~1200

Managed via Puppet almost exclusively

# Puppet in WMF

Some numbers:

# Puppet in WMF

Some numbers:

- 8 git repos containing operations specific code

# Puppet in WMF

Some numbers:

- 8 git repos containing operations specific code
- 50345 lines of Puppet code

# Puppet in WMF

Some numbers:

- 8 git repos containing operations specific code
- 50345 lines of Puppet code
- 53235 lines of ERB (Embedded Ruby)

# Puppet in WMF

Some numbers:

- 8 git repos containing operations specific code
- 50345 lines of Puppet code
- 53235 lines of ERB (Embedded Ruby)
- 50380 lines of Ruby

# Puppet in WMF

Some numbers:

- 8 git repos containing operations specific code
- 50345 lines of Puppet code
- 53235 lines of ERB (Embedded Ruby)
- 50380 lines of Ruby

Numbers are not very accurate and do not account for 3rd party code imported into repos, include tests/RSpec and comments/whitespace.

# State pre Q3 2016

Agents:

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

Puppetmasters:

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

Puppetmasters:

- 2 Precise Pangolin (Ubuntu 12.04) boxes

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

Puppetmasters:

- 2 Precise Pangolin (Ubuntu 12.04) boxes
- 1 ”frontend”, 1 ”backend”

# State pre Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

Puppetmasters:

- 2 Precise Pangolin (Ubuntu 12.04) boxes
- 1 ”frontend”, 1 ”backend”
- Puppet versions 3.4

# State pre Q3 2016

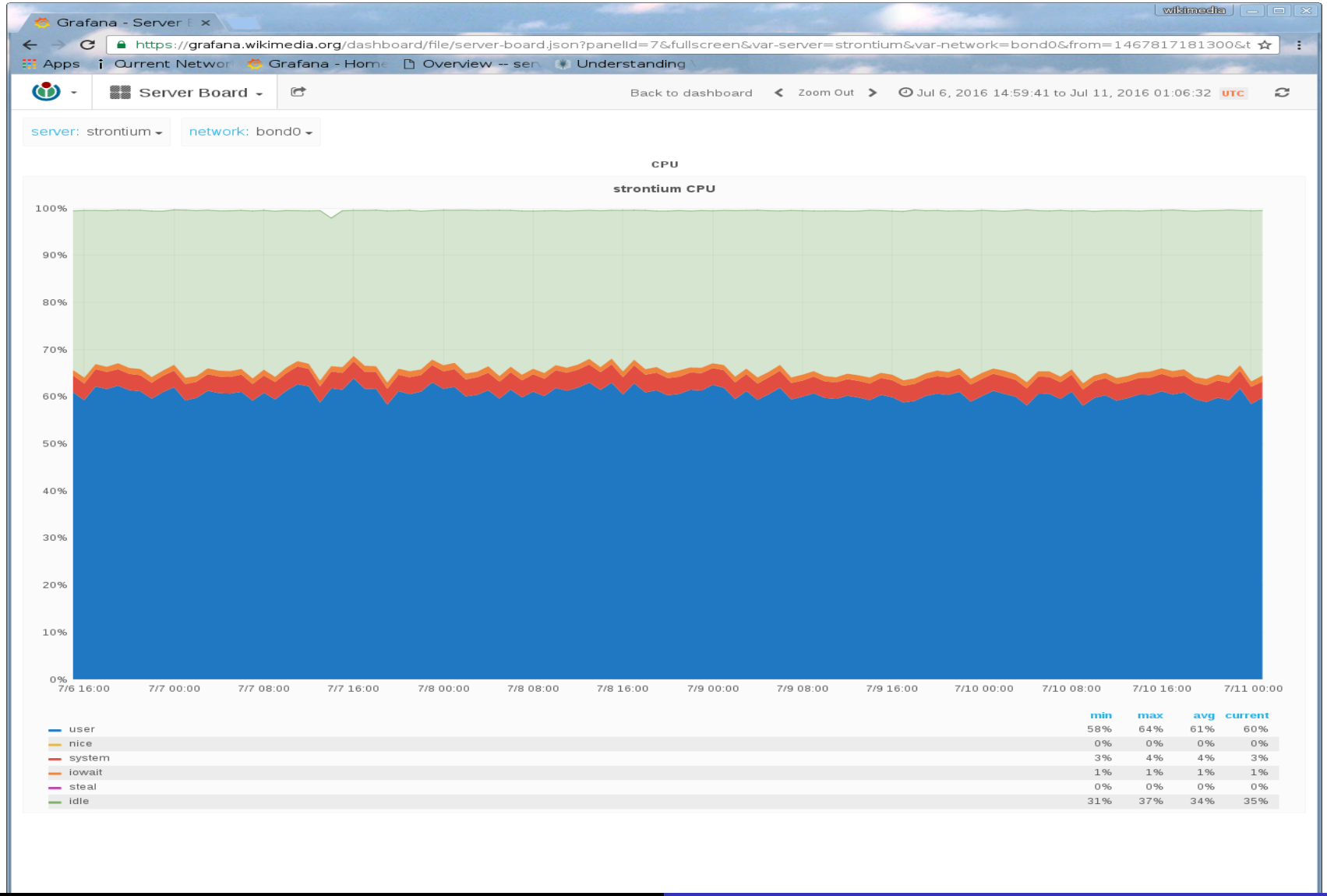
Agents:

- Run every 30 mins with a randomization in agent run start times
- Puppet versions vary from 3.4 to 3.7
- Ruby versions vary from 1.8 to 2.1

Puppetmasters:

- 2 Precise Pangolin (Ubuntu 12.04) boxes
- 1 ”frontend”, 1 ”backend”
- Puppet versions 3.4
- Ruby versions 1.8

# Puppetmaster CPU usage pre Q3 2016



# State post Q3 2016

Agents:

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

- 5 Debian Jessie machines (Debian 8), Ruby version 2.1

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

- 5 Debian Jessie machines (Debian 8), Ruby version 2.1
- 2 ”frontends”, 3 ”backends”

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

- 5 Debian Jessie machines (Debian 8), Ruby version 2.1
- 2 ”frontends”, 3 ”backends”
- Puppet version 3.8

# State post Q3 2016

Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

- 5 Debian Jessie machines (Debian 8), Ruby version 2.1
- 2 ”frontends”, 3 ”backends”
- Puppet version 3.8
- PuppetDB version 2.3

# State post Q3 2016

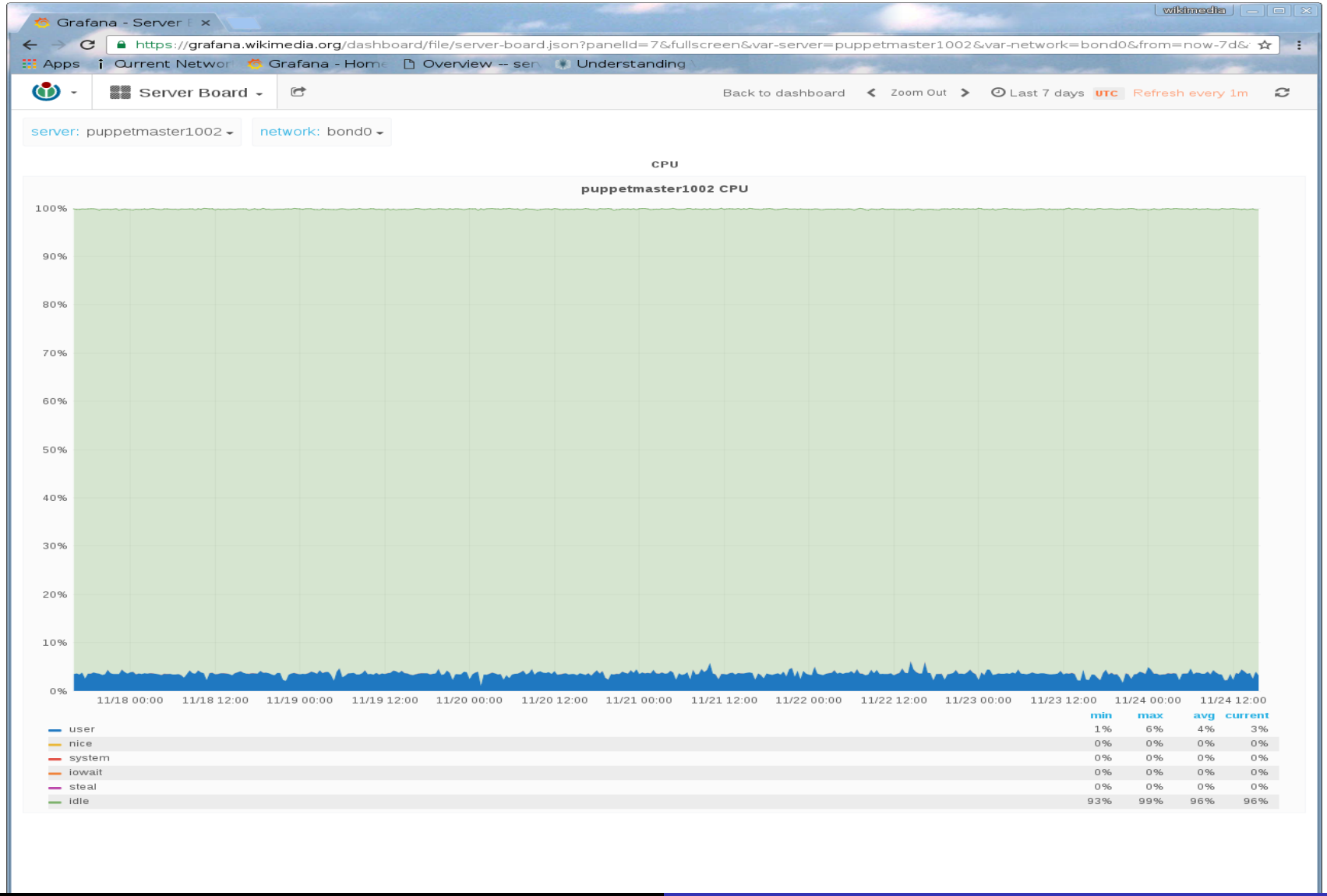
Agents:

- Run every 30 mins with a randomization in agent run start times. Thinking about lowering to 20 though
- Puppet versions vary from 3.4 to 3.8 (but with a lower variance)
- Ruby versions vary from 1.8 to 2.1 (but with a lower variance)

Puppetmasters:

- 5 Debian Jessie machines (Debian 8), Ruby version 2.1
- 2 ”frontends”, 3 ”backends”
- Puppet version 3.8
- PuppetDB version 2.3
- Ruby version 2.1

# Puppetmaster CPU usage post Q3 2016



# Terminology

# Terminology

Backend:

# Terminology

Backend:

- An application server using HTTPS as communication protocol

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

Frontend:

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

Frontend:

- Distributes requests to backends

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

Frontend:

- Distributes requests to backends
- Also an application server since it serves some resources

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

Frontend:

- Distributes requests to backends
- Also an application server since it serves some resources

Storedconfigs:

# Terminology

Backend:

- An application server using HTTPS as communication protocol
- Compiles and serves catalogs and file resources

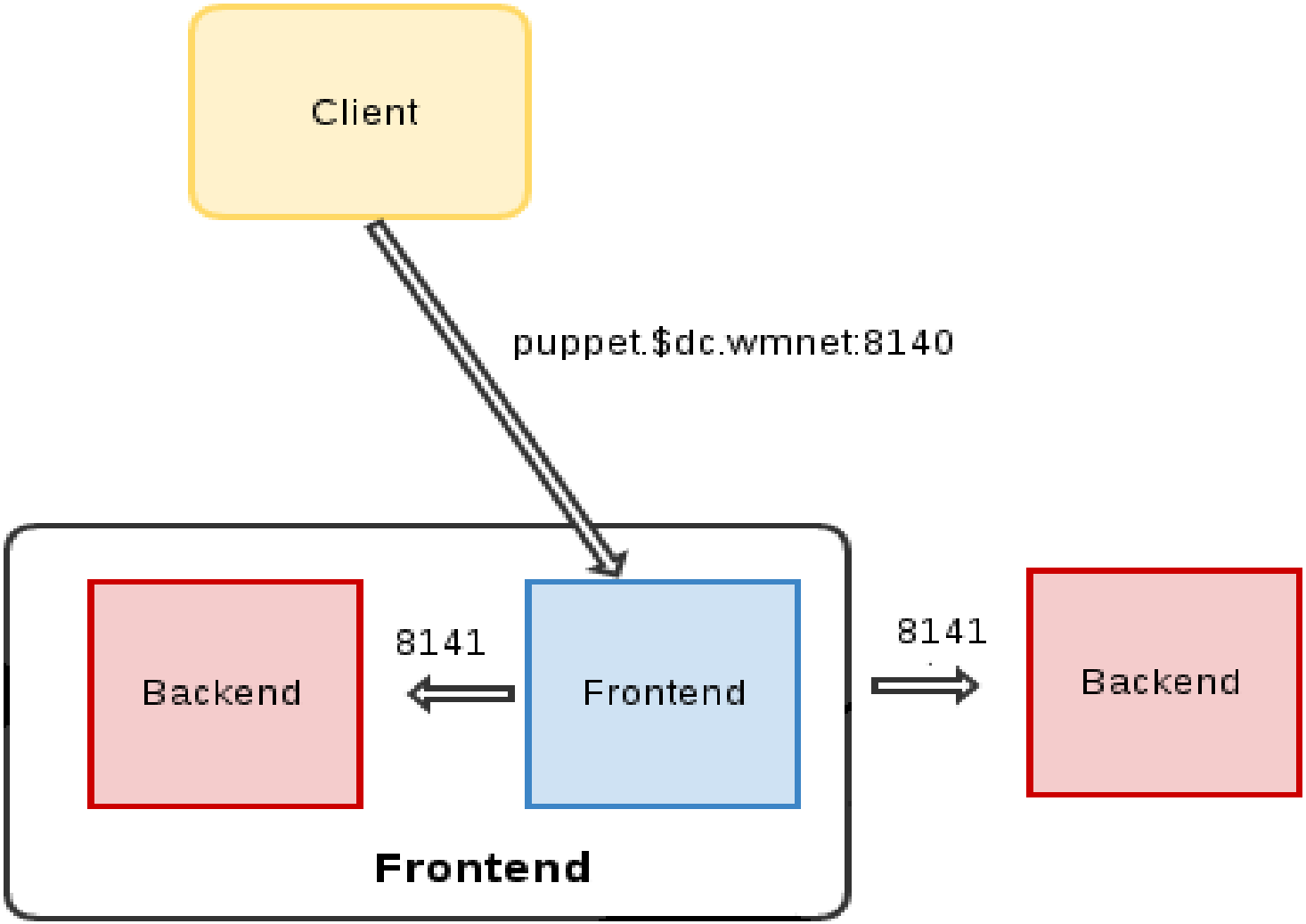
Frontend:

- Distributes requests to backends
- Also an application server since it serves some resources

Storedconfigs:

- A database that stores configuration resources

# Architecture pre Q3 2016



# Architecture pre Q3 2016 (2)

Frontend:

# Architecture pre Q3 2016 (2)

Frontend:

- Apache 2.2 with mod_proxy

# Architecture pre Q3 2016 (2)

Frontend:

- Apache 2.2 with mod_proxy
- Terminates TLS connections and authenticating clients

# Architecture pre Q3 2016 (2)

Frontend:

- Apache 2.2 with mod_proxy
- Terminates TLS connections and authenticating clients
- Reverse Proxy with weighted round-robin to backends (TLS encrypted)

# Architecture pre Q3 2016 (2)

Frontend:

- Apache 2.2 with mod_proxy
- Terminates TLS connections and authenticating clients
- Reverse Proxy with weighted round-robin to backends (TLS encrypted)
- Serves ca, volatile and stores reports, filebuckets exclusively

# Architecture pre Q3 2016 (3)

Backend:

# Architecture pre Q3 2016 (3)

Backend:

- Apache 2.2 with mod_passenger

# Architecture pre Q3 2016 (3)

Backend:

- Apache 2.2 with mod_passenger
- Does the catalog compilations and serves catalogs files

# Architecture pre Q3 2016 (3)

Backend:

- Apache 2.2 with mod_passenger
- Does the catalog compilations and serves catalogs files

Storedconfigs:

# Architecture pre Q3 2016 (3)

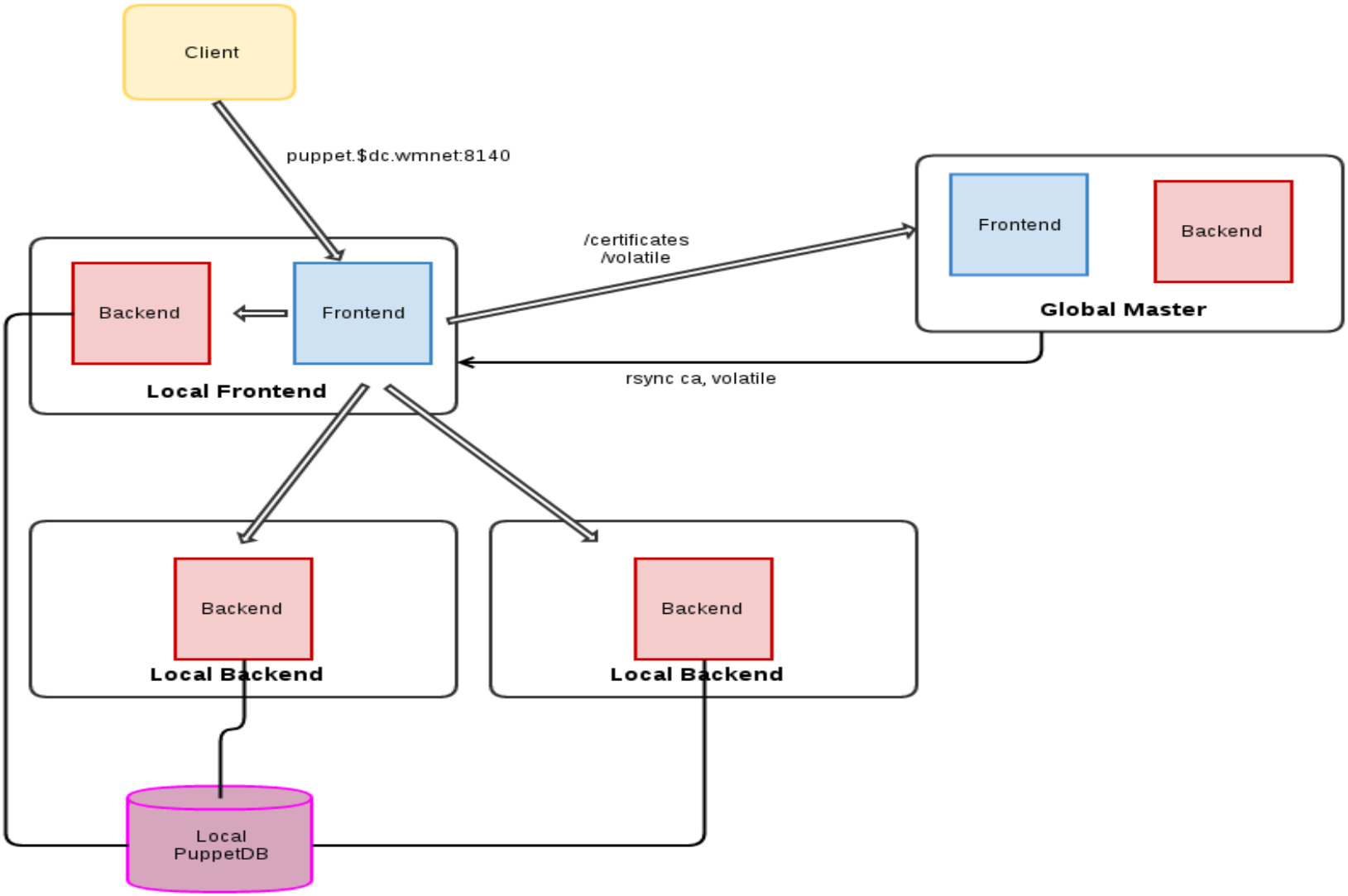
Backend:

- Apache 2.2 with mod_passenger
- Does the catalog compilations and serves catalogs files

Storedconfigs:

- A MySQL database, Active Record Ruby Driver communicating with it

# Architecture post Q3 2016



# Architecture post Q3 2016 (2)

No big changes:

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2
- Agents now contact closest DC ”frontend”

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2
- Agents now contact closest DC ”frontend”
- There is now a ”Master” frontend for some resources

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2
- Agents now contact closest DC ”frontend”
- There is now a ”Master” frontend for some resources
- queries for CA, volatile go to ”Master” ”frontend”

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2
- Agents now contact closest DC ”frontend”
- There is now a ”Master” frontend for some resources
- queries for CA, volatile go to ”Master” ”frontend”
- ca, volatile is being rsynced on frontends (unidirectional )

# Architecture post Q3 2016 (2)

No big changes:

- Apache 2.4 instead of 2.2
- Agents now contact closest DC ”frontend”
- There is now a ”Master” frontend for some resources
- queries for CA, volatile go to ”Master” ”frontend”
- ca, volatile is being rsynced on frontends (unidirectional )
- Backends query puppetDB instead of MySQL for Storedconfigs

# Getting codebase up to date with puppet 3.8, ruby 2.1

# Getting codebase up to date with puppet 3.8, ruby 2.1

- Fleet wide catalog compilation on a puppet compiler

# Getting codebase up to date with puppet 3.8, ruby 2.1

- Fleet wide catalog compilation on a puppet compiler
- Remove all UTF-8 characters

# Getting codebase up to date with puppet 3.8, ruby 2.1

- Fleet wide catalog compilation on a puppet compiler
- Remove all UTF-8 characters
- Strings objects handled differently in Ruby 1.9+

# Getting codebase up to date with puppet 3.8, ruby 2.1

- Fleet wide catalog compilation on a puppet compiler
- Remove all UTF-8 characters
- Strings objects handled differently in Ruby 1.9+
- Hash mutability is now deprecated and discouraged

# Getting codebase up to date with puppet 3.8, ruby 2.1

- Fleet wide catalog compilation on a puppet compiler
- Remove all UTF-8 characters
- Strings objects handled differently in Ruby 1.9+
- Hash mutability is now deprecated and discouraged
- Had to repuppetize a few stuff that got erroded over time

# Get a backend puppetmaster running 3.8

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only
- Switch selectively agents to use the new Virtualhost via `server =` directive in puppet.conf

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only
- Switch selectively agents to use the new Virtualhost via `server =` directive in puppet.conf
- Enable `always_cache_features = true`, otherwise agent runs take +200% time

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only
- Switch selectively agents to use the new Virtualhost via `server =` directive in puppet.conf
- Enable `always_cache_features = true`, otherwise agent runs take +200% time
- Find out a performance issue that caused catalog bloat by 1500%

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only
- Switch selectively agents to use the new Virtualhost via `server =` directive in puppet.conf
- Enable `always_cache_features = true`, otherwise agent runs take +200% time
- Find out a performance issue that caused catalog bloat by 1500%

Marvel at the fact the new backend can serve all the traffic

# Get a backend puppetmaster running 3.8

- New apache Virtualhost on the frontend, new backend only
- Switch selectively agents to use the new Virtualhost via `server =` directive in puppet.conf
- Enable `always_cache_features = true`, otherwise agent runs take +200% time
- Find out a performance issue that caused catalog bloat by 1500%

Marvel at the fact the new backend can serve all the traffic
See the old backend commiting harakiri right after that

# Get the frontend puppetmaster running 3.8

# Get the frontend puppetmaster running 3.8

- A new frontend running Debian Jessie, puppet 3.8, ruby 2.1

# Get the frontend puppetmaster running 3.8

- A new frontend running Debian Jessie, puppet 3.8, ruby 2.1
- Add a new DNS name

# Get the frontend puppetmaster running 3.8

- A new frontend running Debian Jessie, puppet 3.8, ruby 2.1
- Add a new DNS name
- Use once more `server =` to selectively switch agents to use it

# Get the frontend puppetmaster running 3.8

- A new frontend running Debian Jessie, puppet 3.8, ruby 2.1
- Add a new DNS name
- Use once more `server =` to selectively switch agents to use it
- Handle a few more small issues (e.g. TLS key size)

# Get the frontend puppetmaster running 3.8

- A new frontend running Debian Jessie, puppet 3.8, ruby 2.1
- Add a new DNS name
- Use once more `server =` to selectively switch agents to use it
- Handle a few more small issues (e.g. TLS key size)

Replicate the setup in second DC

# Start pointing machines to per DC puppetmasters

# Start pointing machines to per DC puppetmasters

- Decide to use DNS SRV records for puppetmaster selection.

# Start pointing machines to per DC puppetmasters

- Decide to use DNS SRV records for puppetmaster selection.
- Backtrack full speed! 1000+ DNS queries by the agents, 1000% increase in agent run times in remote DCs

# Start pointing machines to per DC puppetmasters

- Decide to use DNS SRV records for puppetmaster selection.
- Backtrack full speed! 1000+ DNS queries by the agents, 1000% increase in agent run times in remote DCs
- Go with standard DNS A records

# Start pointing machines to per DC puppetmasters

- Decide to use DNS SRV records for puppetmaster selection.
- Backtrack full speed! 1000+ DNS queries by the agents, 1000% increase in agent run times in remote DCs
- Go with standard DNS A records

Feel happy!

# Start pointing machines to per DC puppetmasters

- Decide to use DNS SRV records for puppetmaster selection.
- Backtrack full speed! 1000+ DNS queries by the agents, 1000% increase in agent run times in remote DCs
- Go with standard DNS A records

Feel happy! Add AAAA records! Feel even happier!

# Get puppetDB running

# Get puppetDB running

- Setup up master/slave Postgres in Primary/Backup DCs

# Get puppetDB running

- Setup up master/slave Postgres in Primary/Backup DCs
- Compile/Package PuppetDB. Fail, use the vendor packages

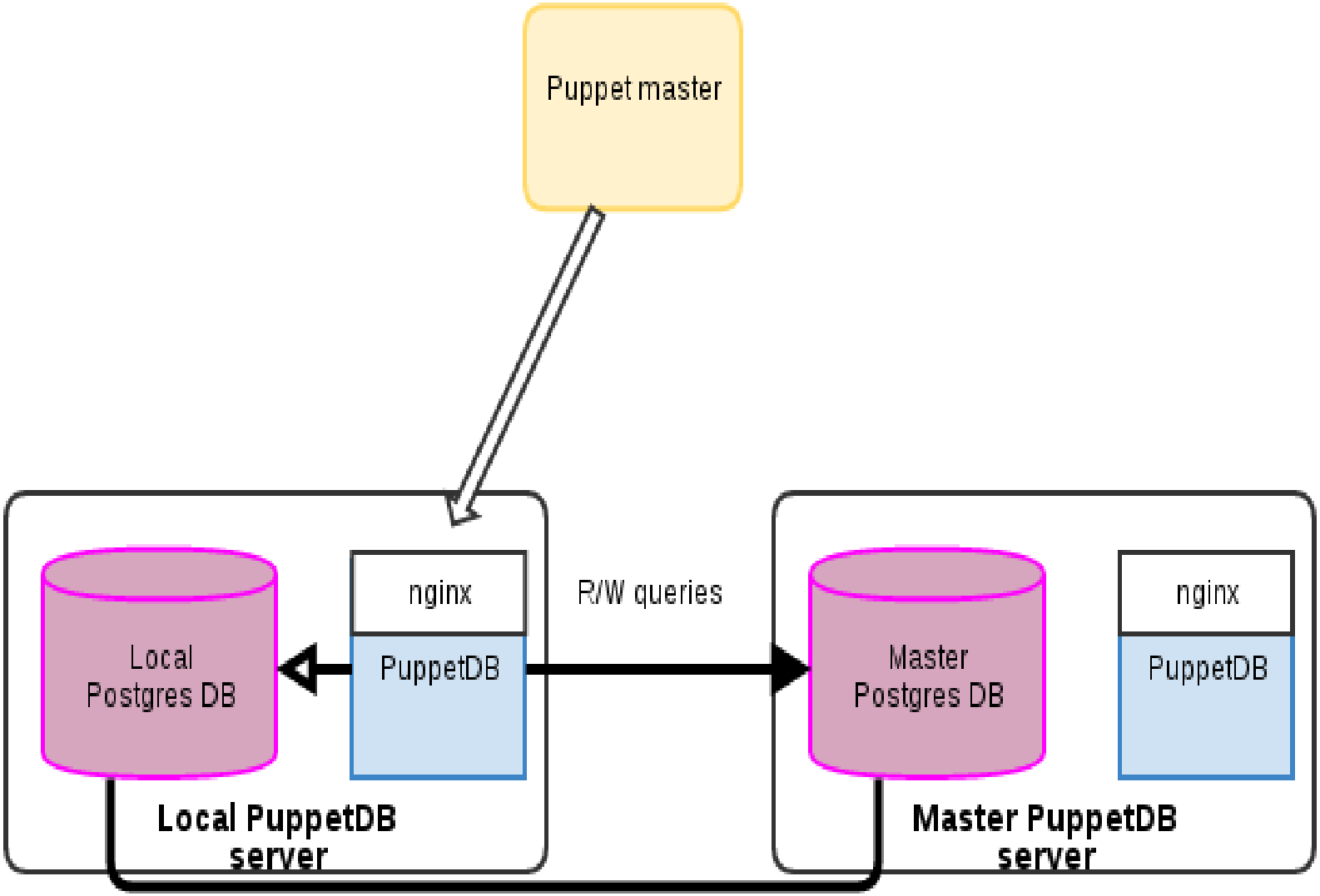
# Get puppetDB running

- Setup up master/slave Postgres in Primary/Backup DCs
- Compile/Package PuppetDB. Fail, use the vendor packages
- nginx for TLS termination proxies to PuppetDBs

# Get puppetDB running

- Setup up master/slave Postgres in Primary/Backup DCs
- Compile/Package PuppetDB. Fail, use the vendor packages
- nginx for TLS termination proxies to PuppetDBs
- Read queries DC local, write queries go to active DC

# Architecture post Q3 2016 (PuppetDB)



# Use/tune puppetDB

# Use/tune puppetDB

- Remove reports storing

# Use/tune puppetDB

- Remove reports storing
- Set Garbage Collection policies

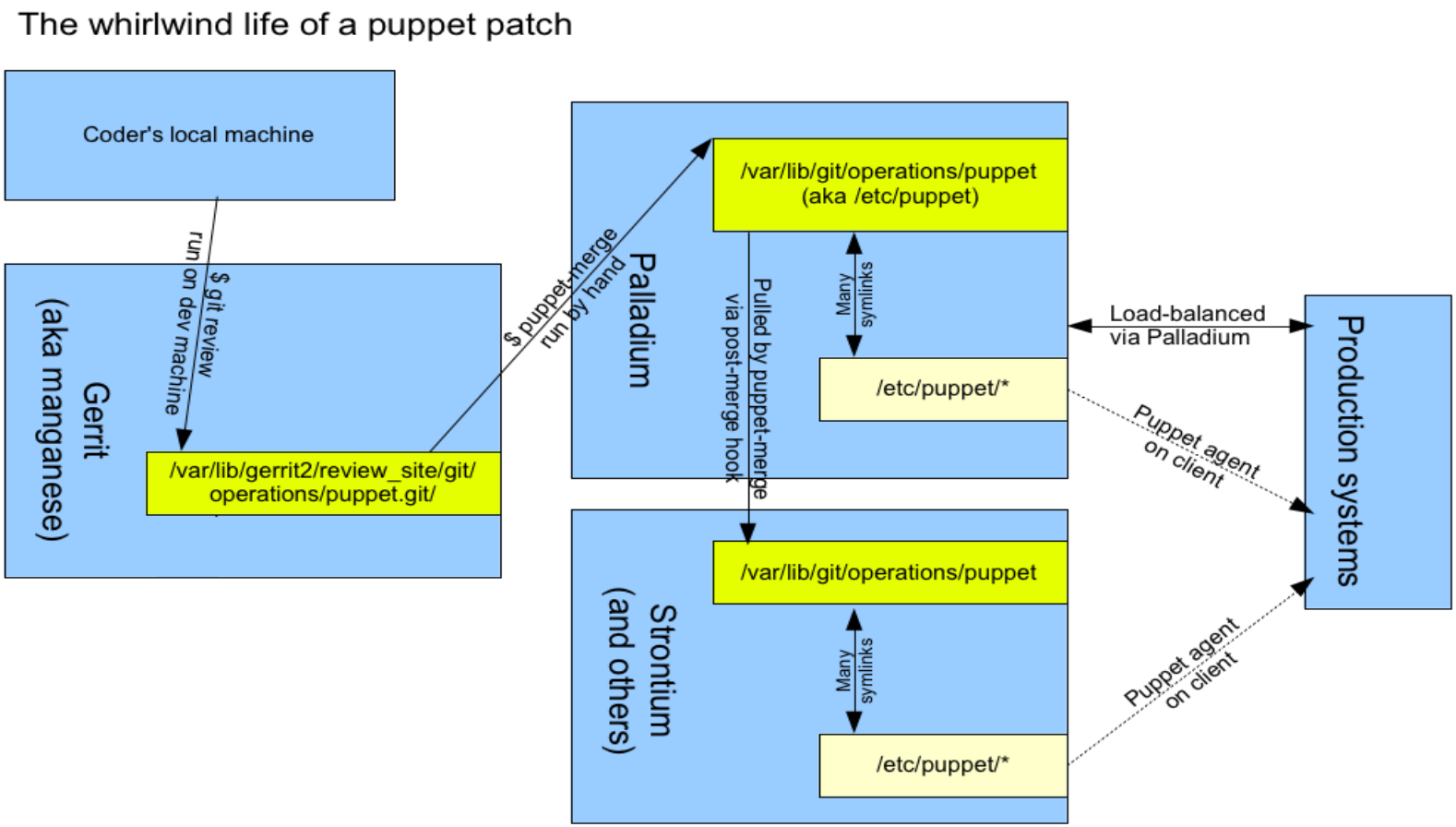
# Use/tune puppetDB

- Remove reports storing
- Set Garbage Collection policies
- Update scripts to start using it. Ugly ugly syntax

# Use/tune puppetDB

- Remove reports storing
- Set Garbage Collection policies
- Update scripts to start using it. Ugly ugly syntax
- Marvel at the WONTFIX bugs. Still working on them

# How code gets updated



# Conclusions

# Conclusions

- Multi DC puppetmasters definitely worth it

# Conclusions

- Multi DC puppetmasters definitely worth it
- PuppetDB can be called production ready

# Conclusions

- Multi DC puppetmasters definitely worth it
- PuppetDB can be called production ready
- Postgres replication is limiting

# Conclusions

- Multi DC puppetmasters definitely worth it
- PuppetDB can be called production ready
- Postgres replication is limiting
- Read the changelogs many many times over

# Links

- https://github.com/akosiaris/presentations
- https://github.com/wikimedia/operations-puppet
- https://github.com/wikimedia/operations-software-puppet-compiler

# Questions

Questions ?