

TECHNOLOGY DEPT

Foundational & Sustaining Programs

May 2018 quarterly check-in
for work done in Q3 FY2017/18



WIKIMEDIA
FOUNDATION

Program Structure

Sustaining	TP1 Availability, Performance & Maintenance	TP3 Addressing Technical Debt TP8 Multi-datacenter Support
Foundational	TP2 MediaWiki Refresh TP6 Streamlined Service Delivery PP1 Discoverability	X-SPDM Security, Privacy, & Data mgmt X-SDC Structured Data on Commons
Community Support	TP5 Scoring Platform (ORES) TP9 Growing Wikipedia Across Languages TP7 Smart Tools for Better Data	TP11 Citations/Verifiability TP12 Growing Contributor Diversity X-CH Community Health/Anti-harassment
Tech Community Support	TP4 Technical Community Building	TP10 Public Cloud Services & Support

Program Priorities

	If we don't do this...	Actual FTE (approximate)
Sustaining	The sites go down.	30
Foundational	Performance and data quality decays.	10
Community Support	Become technologically obsolete.	10
Tech Community Support	Lose bots and code contributions.	4

How we prioritize



Fundamentals

What is our part in fulfilling the mission?



Service

What are other people asking from us?



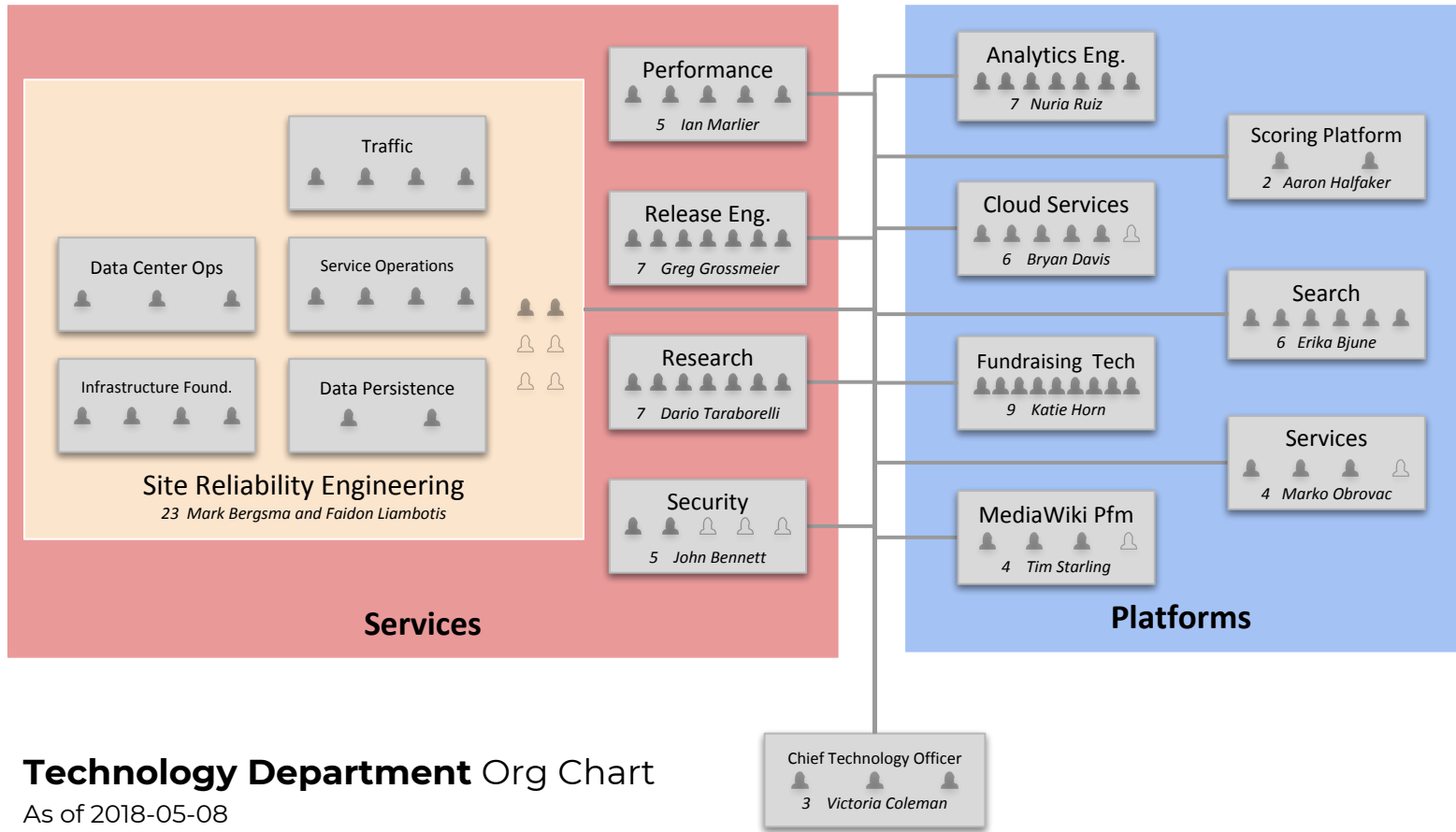
Improvement

What could we do to improve our offering?



Maintenance

What will sustain and improve our delivery?



Technology Department Org Chart

As of 2018-05-08

Includes filled and vacant Reqs only

Agenda:

Technology

- Program 1: Availability, performance and maintenance
- Program 2: MediaWiki
- Program 3: Addressing technical debt
- Program 6: Streamlined service delivery
- Program 8: Multi-datacenter support

Product / Audiences & Cross-Departmental (tech portions only)

- Program 1: Make knowledge more easily discoverable
- Segment 2: Security Update

Programs covered in other presentations:

- Product Program 7: Payment processor investigation and long-term strategy
- Product Program 8: Donor retention
- Fundraising Tech / Advancement
- Privacy, Security and Data Management
- Structured Data on Commons (Database & Search integration)

Technology Program 1

Availability, Performance and Maintenance

Program Structure

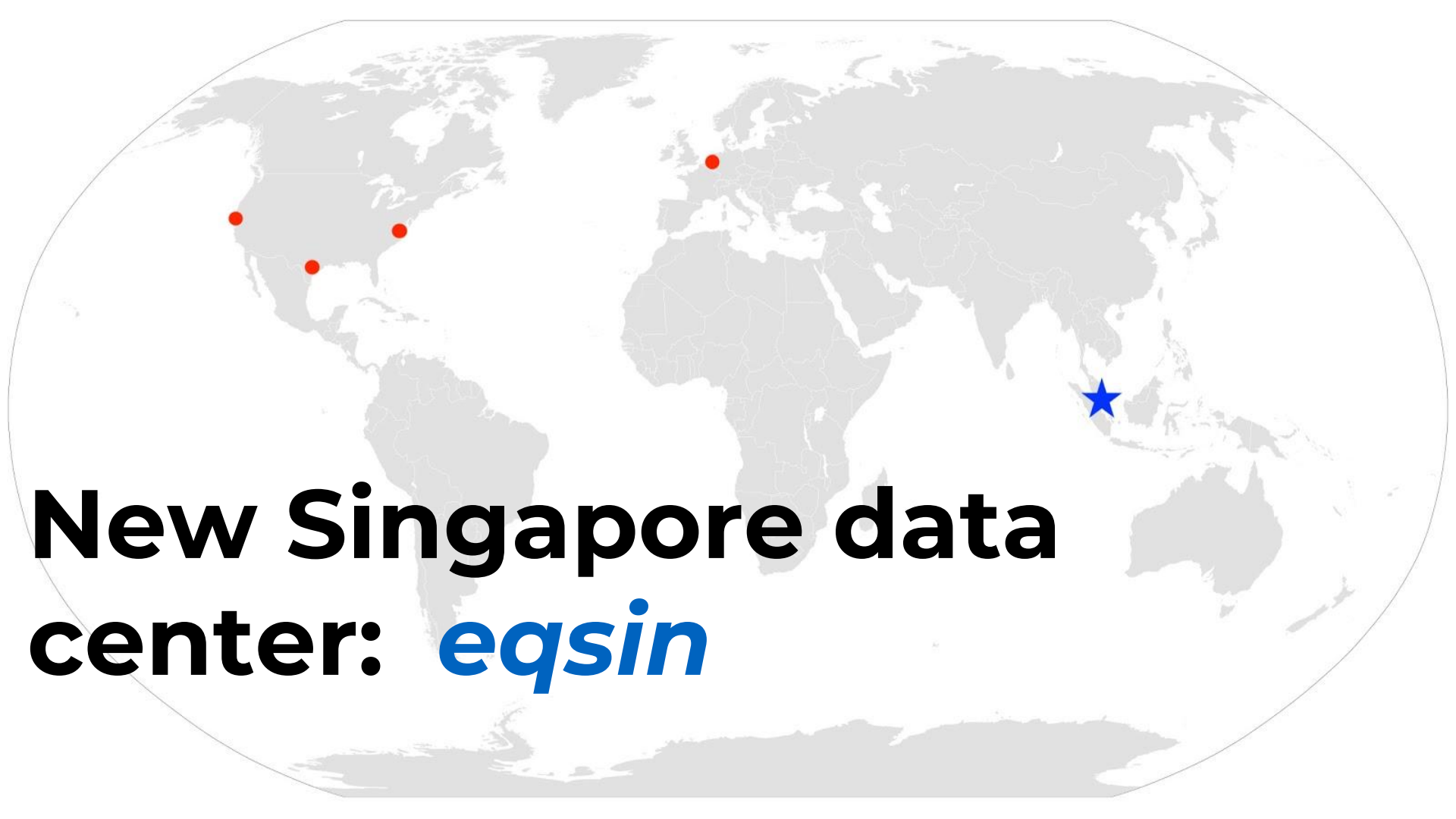
Sustaining

TP1 Availability, Performance & Maintenance

We will **maintain** the *availability* of Wikimedia's sites and services for our global audiences, and ensure they're **reliable**, **secure** and **performant** while *modernizing* our infrastructure and improving current levels of service when it comes to testing, deployments and maintenance of software and hardware



Scaffolding of Big Ben for maintenance, by David Holt, CC-BY-SA 2.0



**New Singapore data
center: *eqsin***

The new **cache edge site** now serves the Asia-Pacific region:

- Procured **3x diverse transit links** and a **transport backhaul** to codfw
- Established peering with regional networks
- Optimized routing to most of the region
- **Enabled live service** for a limited subset of the region by end of Q3
 - ...most of the rest of the region shortly afterwards
- Initial performance metrics gathered
 - detailed in [public blog post](#)
 - **Page load times from Indonesia and Bangladesh dropped by ~33%**
- Q4 ongoing work:
 - Further optimization (*e.g. China and Australia transit issues*)
 - Trailing Zero issues (*e.g. Thailand, Myanmar, East Timor*)

Status: Done

Edge cache clusters upgraded



Image from Internet Archive (free use)

- **Varnish** major version upgrade from **4.x to 5.x**
- Both larger clusters upgraded (*text, upload*)
 - Smallest cluster (*misc*) done in the previous quarter
- **Handles all of our average 100K reqs/sec** to all Wikimedia projects/sites from the Internet.
- **Answers 91% of reqs directly** without sending requests to other layers of our Tech stack.
- Upgrade benefits:
 - Alleviated some scaling issues in upload cluster
 - Enables future work to reduce cluster count
 - Allows tracking upstream patches closely

Status: Done



Database backups

After issues with **database backup** generation infrastructure — we made *backups* a focus.

In **Q3**, we deployed *redundant* database backup generation servers in our secondary data center **codfw**, to ensure *consistent* logical backups and *exact* backup data consistency...

On **Feb 6**, the old (*still non-redundant*) **eqiad** db backup generation server **crashed**...

...throwing a spanner in the wheel of this goal.

Status: Done anyway



Puppet upgrades

Puppet is our primary Configuration Management/IaC software for all servers.

The Puppet master infrastructure was upgraded to **Debian stretch** and **PuppetDB** was upgraded to v4.4. A major and complicated upgrade that also required Puppet code changes and modifications to Cumin.

An initial evaluation of a **future Puppet 5 upgrade** was completed and will follow in the next fiscal year.

Status: Done



Puppetboard

Overview

Nodes

Facts

Reports

Metrics

Inventory

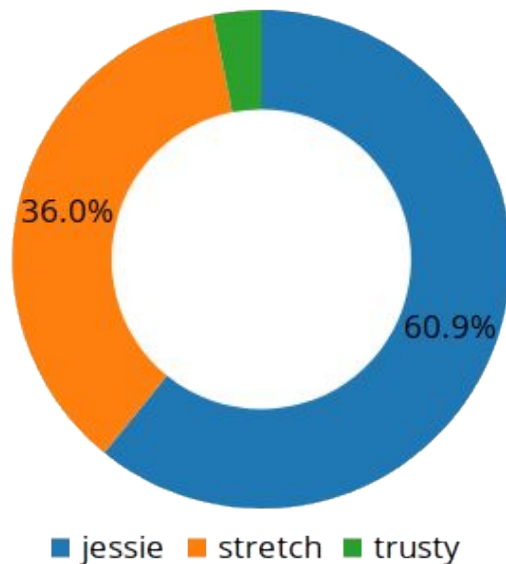
Catalogs

Radiator

Query

Multiple Puppet dashboards were evaluated

Puppetboard was selected and deployed.



HW refresh & expansions

Goal: address a growing backlog of **hardware refreshes** and execute Q3 refreshes, expansions and decoms.

Out of the **13 migrations** listed in this goal, **11** were completed by the end of Q3, the other two will be finalized in Q4.

A separate goal was listed to **cleanup and refresh** our **esams** data center (Haarlem, NL). *This goal did not materialize due to time constraints.*

Status: partially done



Analytics

This probably went unnoticed cause it happened without a glitch ...

We upgraded



to Java8 and added 8 new nodes
for a **new total of 50 nodes**

We are moving analytics clients to a new cluster



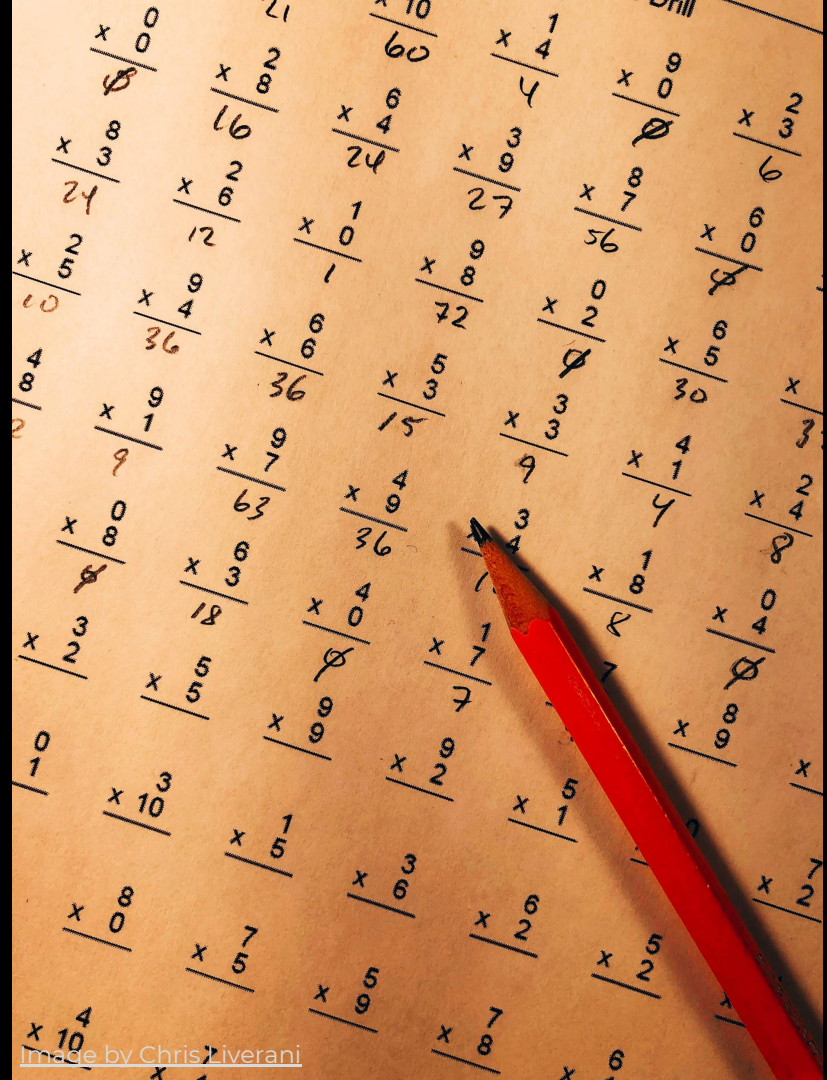
...enabling us to simplify the configuration and management of encryption for cross data center communications

Release Engineering

Develop and migrate to a JavaScript-based browser testing stack

Completed almost all planned
improvements to the new NodeJS
browser test framework:

- Video recording of browser tests almost ready and waiting to be merged
- Upgraded webdriverIO to version 4.9
- Replaced/upgraded the API wrapper (*nodemw* with *mwbot*)



Release Engineering

Maintain existing shared *Continuous Integration Infrastructure*

- Unified how Docker images are built in production and CI
- Drafted basic requirements for Kubernetes cluster for CI
- Migrated “long tail” of CI jobs off Nodepool

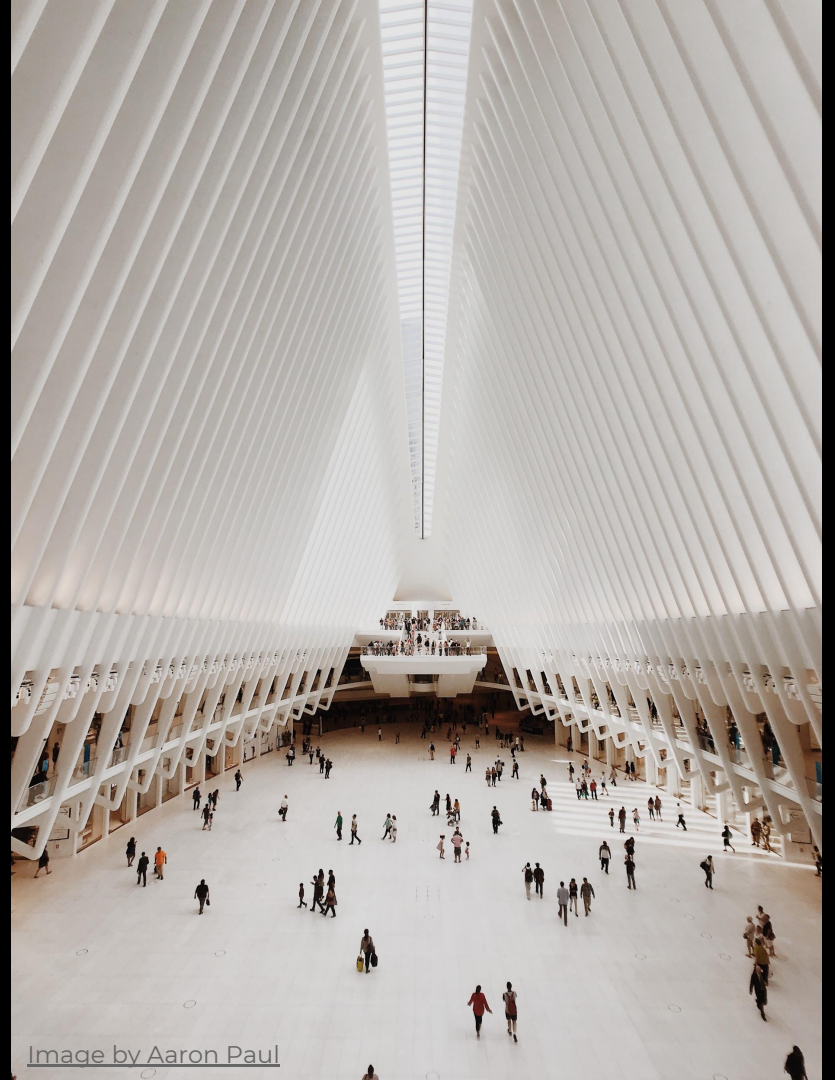


Image by Aaron Paul

Technology Program 2

MediaWiki

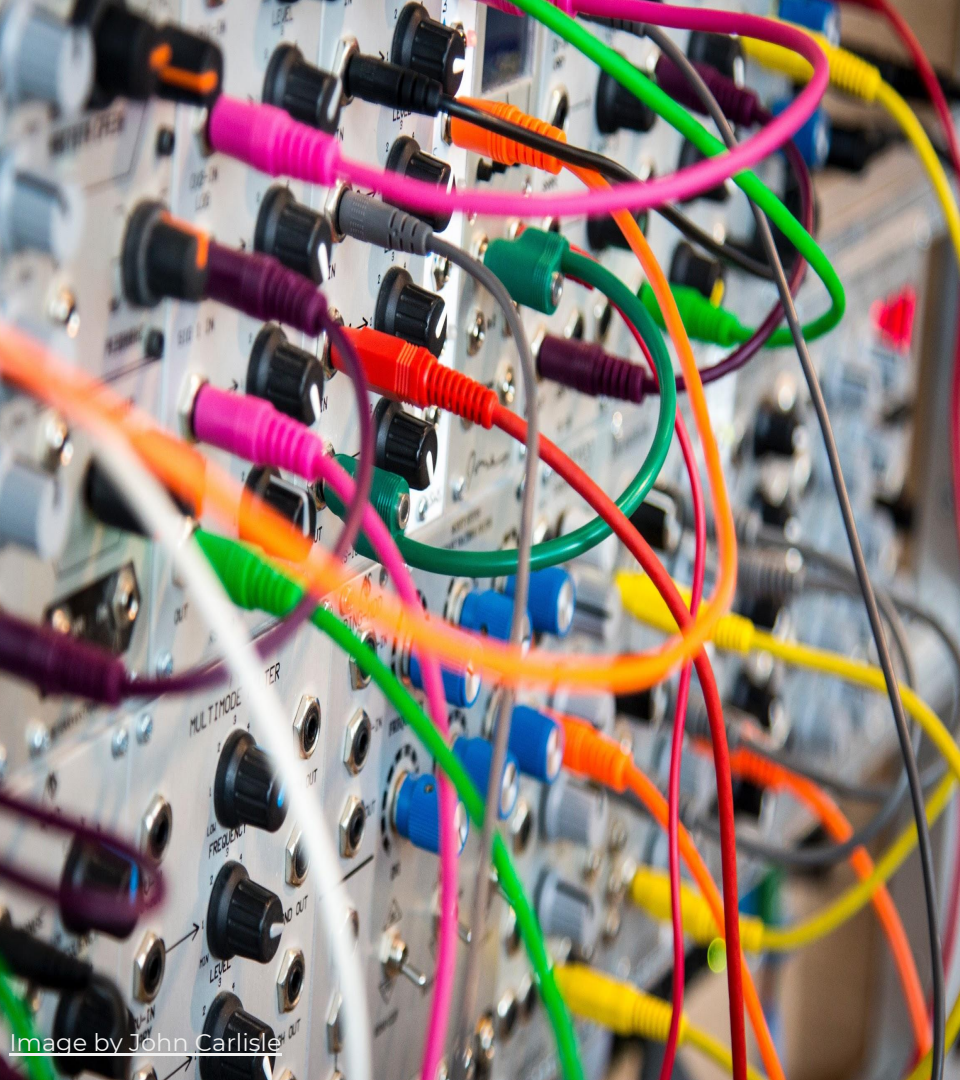
Program Structure

Foundational

TP2 Mediawiki Refresh
TP6 Streamlined Service Delivery
PP1 Discoverability

We will strive for a refreshed and highly performant core platform by bringing renewed focus on MediaWiki





How can we reduce complexity?

Created a plan for the evolution of the technology stack:

Platform Evolution CDP

- annual plan goals
- 3-5 year plan

Status: Complete

How can we increase code quality?

Improvements to:

- continuous integration
- code coverage analysis
- extension management

Core and extension bug fixes

Code review

Status: On-going

Security and Stability

Firejail:

- a sandbox for external commands run by MediaWiki core
- integrated into MediaWiki core
- contributed new features to Firejail upstream maintainers

Phan-Taint:

- static analysis to find security vulnerabilities in MediaWiki extensions

Status: Complete



Security and Stability

Remove use of PHP serialization in revision storage:

- avoid arbitrary execution of malicious code injected into database

Status: Code written and in review, target completion end of Q4

Technology Program 3

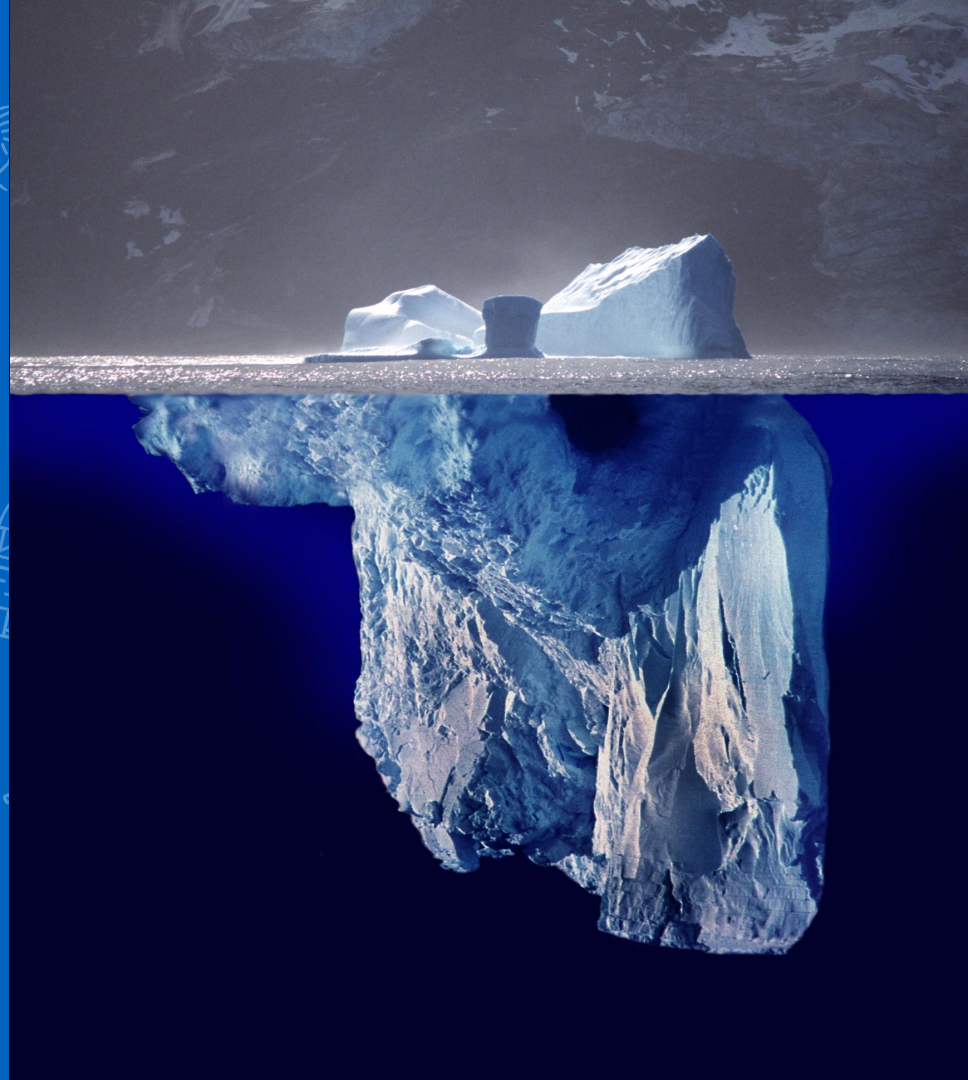
Addressing technical debt

Program Structure

Sustaining

TP3 Addressing Technical Debt
TP8 Multi-datacenter Support

We will lead an organization-wide effort to **reduce technical debt**



Rel-Eng Tech Debt

Implemented the Code Stewardship Review process

- Reviewed and assigned stewards for four areas.
- Q4 review cycle started, with five review candidates.

Defined Technical Debt Engagement Model

- Approach to engaging with development teams to assess and plan technical debt reduction.
- Two teams identified for Q4 engagements.

Technical Debt/Code Health Awareness Campaign

- Regular communication on the topics of Code Health and Technical Debt.
- May 2018 Hack-a-thon session, quarterly newsletters, monthly SIG.

Status: Done

Technology Program 6

Streamlined service delivery

Program Structure

Foundational

TP2 MediaWiki Refresh
TP6 Streamlined Service Delivery
PP1 Discoverability



We will **streamline** and *integrate* the delivery of services, by building a **new production platform** for integrated development, testing, deployment and hosting of applications



SRE

After **designing** and **building infrastructure** for over a year, we deployed the first service *in production* on the Kubernetes cluster.

By the end of the quarter, **100%** of Mathoid traffic (*above the goal of 50%*) was successfully being served by our Streamlined Service Delivery platform.

This validated the design of our **standardized application environment** to work as expected.

Status: Done

Release Engineering

A working POC for the **test, staging, and production deployment pipeline** was successfully built for the mathoid service.

This **verifies the functionality of all the pieces** to provide production image building and deployment.

Status: Done

Services

GOAL: Develop tools allowing users to create, manage and test their services in minikube

Due to changes in the team and shifting priorities, this goal was not tackled in Q3.

- When Gabriel left in late 2017, the team was left with 3 FTEs for 3 goals, two of which (*storage redesign and JobQueue migration*) were high-priority and intensive goals.

Status: Not Done

Technology Program 8

Multi-datacenter support

Program Structure

Sustaining

TP3 Addressing Technical Debt
TP8 Multi-datacenter Support

Program 8 Multi-datacenter support

- 1. Fast primary datacenter switchover**
- 2. Dual primary data centers**

SRE

In Q1, MediaWiki **EtcdConfig** was finished:

Integration of MediaWiki's configuration with **etcd** (a service discovery daemon) for **automatic & instant** configuration changes. This is essential for speedy data center switchovers, as many configurations need to be changed quickly during failovers.

In Q3 we validated EtcdConfig to work well in different **failure scenarios**.

Conftool was extended to allow *safe* automatic editing of MediaWiki configuration values.

EtcdConfig was then successfully **deployed in production**.

Status: Done



Performance

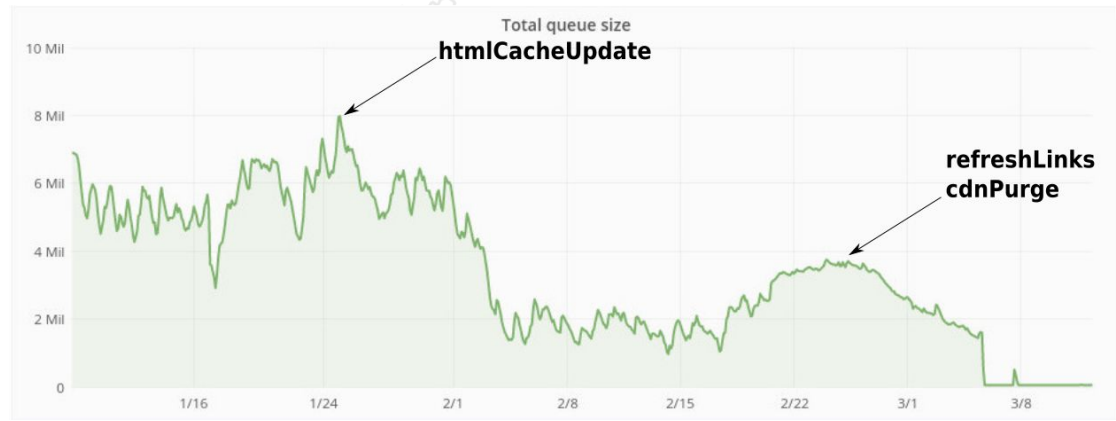
mcrouter deployment

- A Facebook product which provides memcached replication.
- Using mcrouter allows us to have a “hot” cache in both data centers
 - this means that services will not be affected when we fail over from one data center to the other.
- Goal: install and use in deployment-prep (Beta Cluster)
- In progress at end of Q3

Status: Done (completed in Q4)

Services

Migrate at least **two**
high-traffic jobs to the
new infrastructure



High-traffic jobs have been causing big scalability problems in our infrastructure.

In Q3, **we migrated the three most-high-traffic jobs**: refreshLinks, htmlCacheUpdate and cdnPurge.

Many code changes went into this effort: changes to the refreshLinks code, allowing and enabling the processing of multiple partitions (due to MySQL scalability issues), improved de-duplication, etc.

Status: Done

Product Program 1

Make knowledge more easily discoverable

Program Structure

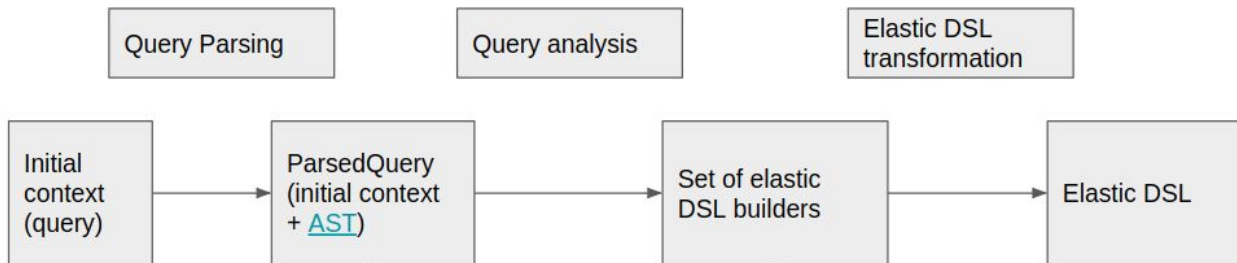
Foundational

TP2 Mediawiki Refresh
TP6 Streamlined Service Delivery
PP1 Discoverability

Objective 1:

Implement advanced methodologies to improve search result relevance across language Wikipedias

- **In progress:** Implemented **profile management in CirrusSearch** ([T183279](#)) and started work on revamping the Cirrus query parser as part of a large and [ongoing refactor plan](#) ([T185108](#))

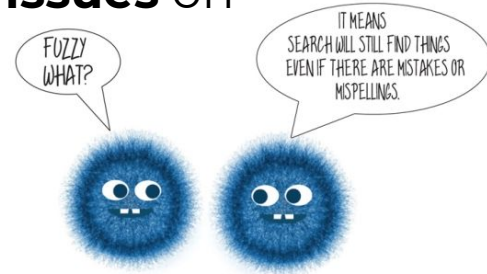


- **Done: Evaluated query explorer** feature in the Learning-to-Rank plugin for machine learning ([T187148](#)) and deployed new models to 19 wikis

Objective 2:

Improve support for multiple languages by researching and deploying new language analyzers

- **Done: Phonetic/fuzzy search investigation** completed ([T182708](#))
- **In progress:** Continue to investigate morphological libraries for Elasticsearch plugins
 - **Done: Serbian plugin and analysis chain** implemented ([T183015](#)) and **Slovak analyzers investigated** ([T178929](#))
- **Done: Addressed Cyrillic-to-Latin transliteration issues** on Crimean Tatar language wiki ([T186727](#))
see Trey's blog post on [supporting languages with multiple writing systems](#)



Objective 3:

Investigate how to expand and scale Wikidata Query Service to improve its ability to power features on-wiki for readers

- **Done:** Implemented **access to Categories via WDQS** on select wikis ([T181549](#))
- **Done:** Added **SPARQL client as a service in MW Core** ([T185127](#)) and added **“deepcategory” keyword** to access subcategories ([T184840](#))
- **Done:** Acquired and racked **six new servers for WDQS** ([T178492](#))
- **Done: Indexing of Lexeme lemmas and forms** as first phase of implementing Lexeme search in Wikibase instances ([T189736](#))

Security

Goals in progress:

- Solidify **compliance**
 - Develop protection framework to align with **GDPR** (*done*)
- Create app security training
- Complete one security release
- Ongoing security reviews
- Work with Legal (*and other teams*) to support security and privacy incidents
- Draft Risk Management framework
- **Privacy**: Create and train on WMF PGP keys (*to do in Q4*)



Questions?





Additional work:

Reactive, one-offs and notable achievements not mentioned previously

Site Reliability Engineering

Other achievements

- Finished firewall deployments on all WMF MariaDB installs
- Implemented more Prometheus-based icinga alerting
- Implemented IPv6 in our Kubernetes environment
- Deployed on average 35.7 security package upgrades per server
- Increased PyBal unit test statement coverage from 71% to 90%
- Tracked down MediaWiki API infinite loop HHVM hang [w/many]
- Deployed LibreNMS IRC bot in #wikimedia-operations
- Custom Cumin backend for WMCS
- Built and rolled out 4 HHVM releases
- dbproxies upgraded to stretch/1.7/new kernel
- Published helm chart repo
- Moved ORES to a dedicated cluster
- Tested Dynomite for memcached, turned 180 degrees to Mcrouter
- Implemented Varnishospital to monitor origin server health [w/Performance]
- Replaced ulsfo switches with new models
- Improved performance/parallelism of db data checksumming script
- Implemented candidate masters for all DB primary masters for emergencies
- misc databases upgraded to stretch/new kernel
- Fully decommissioned ganglia.wikimedia.org
- Finished migration from AS43821 to AS14907
- Fixed several critical PyBal bugs in several new release
- Implemented multi-instance on vslow,dump mediawiki dbhosts in some sections

Other achievements

- Started deprecation of legacy Graphite-based collector, Diamond
- Deployed kernel updates to address Spectre and Meltdown vulnerabilities
- Audited external monitoring checks
- Assisted with Thumbor duties: review, deployment and Debian uploads [w/Performance]
- Expanded coverage of standard java JVM Prometheus monitoring
- Two FOSDEM talks (TLS for MySQL & Cumin)
- Implemented smarter Varnish Slow log
- Evaluated Netbox
- Configured redundant frack NICS [w/fr-tech]
- Finished many long-running schema changes
- Upgraded MariaDB/kernel on most databases
- Migrated dumps to labstore1006/1007 [w/WMCS]
- Cleaned up icinga-wm code and increased reliability
- Deployed Annual Report 2017 site [w/Comms]
- Various docker improvements in CI
- Parse Varnish and Nginx request logs into Prometheus metrics
- Made misc-static-sites (15.wp, transparency/annual report) multi-dc
- Held a 2-day mini-onsite in SF
- Three new releases for Cumin, including Python 3 porting
- Automated service restarts for stateless system services
- SMART checking and reporting implemented fleet-wide

Cloud Services



Other achievements

- Migrated wikitech wiki to new servers (first time there has been multiple servers for this wiki), a new operating system (first time on Debian), and a new runtime (first time running on HHVM).
- Migrated wikitech database to 'm5' database cluster.
- Migrated Horizon to new servers (first time on multiple servers) using a new deployment system that will allow us to track upstream changes better (scap3 instead of the former deb packages).
- Migrated Striker to new servers.
- Continued to make progress on OpenStack Neutron software defined networking migration with a working Neutron service in our 'labtest' staging environment.