Technology Group

Architecture, Technical Operations, Release Engineering, Services and Security

FY Q3: January - March 2015/16

Quarterly review

Architecture

Q3 - 2015/16

Approximate team size during this quarter: 1 FTE

Time spent: focus 70%, strengthen 30%

Key performance indicator

None defined			

Q3 - Architecture

Objective: lending ArchCom authority

Objective	Measure of success	Status
Focus: Distributed leadership at/beyond WikiDev '16 (building on momentum from WikiDev '16)	Establish process for forming working groups, with significant progress in forming a first set of groups	Partial. Piloted Rust-inspired process (T123606), but proposals haven't catalyzed formation of working group working under those rules

Rust process involves “subteams”. The word “subteams” triggers fear of the Architecture Committee (ArchCom) overstepping its authority outside of the team’s expertise.

“Shepherds” concept from Rust seems to hold a lot of promise in sustaining forward momentum on proposals. Clarifying official liaison within ArchCom helps proposal authors know who on ArchCom to work with, and helps clarifying commitments from ArchCom members.

Q3 - Architecture

Objective: Improve RFC documentation

Objective	Measure of success	Status
Strengthen:Improve RFC documentation. Better checklists for arriving at engineering consensus (e. g. T118473)	Checklists created, with more understandable process.	Done [[mw:Consensus]] and [[mw: Good meetings]] intended as guides for developers trying to build consensus for their ideas.

Valerie Aurora's lessons on good meeting practice at WikiDev '16 and All Hands seem to have resulted in enduring lessons.

Quarterly review

Operations

Q3 - 2015/16

Approximate team size during this quarter: 17 FTE

Time spent: strengthen 45%, focus 45%, experiment 10%

Key performance indicator

Availability	99.962%	-0.017% from Q2	

Q3 - Technology / Operations

Objective: DR testing

Objective	Measure of success	Status
Focus: Make `codfw` (Dallas data center) functional as a backup data center *Technology team members involved: 15* *Additional team members outside Technology: 4*	Serve MediaWiki application server traffic from `codfw` for 48 hours. Serve Swift, ElasticSearch and RESTBase, Parsoid services from codfw. Ability to deploy MediaWiki, RESTbase, Parsoid from codfw.	Preparation for the 48 hour switchover almost entirely complete by EOQ, and several coupled services were tested. The final test with MediaWiki itself was postponed for various unforeseen circumstances.

Several unexpected obstacles, higher risks and challenges with timing made us decide to move the migration by a few weeks, even though we were able to proceed if absolutely required. The migration will be performed on April 19th instead and we are now better positioned to make this a success, mitigate more risks and reduce impact to our users.

Learning: We should have put more effort in planning/investigating potential dates & communication earlier and realized that the last week of this quarter was suboptimal for this effort. Thanks to Sherry Snyder for the (unplanned) communications support.

Q3 - Operations

Objective: Labs dashboard

Objective	**Measure of success**	**Status**
Strengthen: Move basic Labs administration workflow from wikitech to Horizon *Team members involved (+outside Ops): 2 (+1)*	2FA Horizon authentication. Support instance manipulation (create/delete). Allow modification of Labs public DNS records. Allow manipulation of HTTP Proxies.	Achieved all technical aspects, and users are currently using Horizon for real work. DNS and proxy management are exclusively in Horizon now.

Learning: We benefited a lot from Alex Monk's involvement (of the VisualEditor team, thanks!) with customizing Horizon, especially as he wrote most of our UI for proxy management. Chris Steipp (Security) was involved in the 2fa rollout for both code and review.

The new DNS management has a stricter security model in which each domain is owned by a specific project. Existing uses that violate this rule have been grandfathered in but cannot be manipulated using the new web UI.

Q3 - Operations

Objective: Maintenance tech debt

Objective	Measure of success	Status
Strengthen: Eliminate Ubuntu 12.04 Precise systems in our infrastructure (technical debt) *Team members involved: 7 (+1)*	Reduce amount of remaining Ubuntu 12.04 systems from 100 to <= 40.	Over 60 systems were migrated to Debian Jessie, for a final count of 36 Ubuntu 12.04 systems remaining at EOQ.

The reduction and eventual elimination of older, now deprecated operating system distributions (and corresponding customly maintained packages) significantly reduces our maintenance burden. However, some types of systems (e.g. core database servers) require a lengthy, careful migration process which will continue into the next quarter(s).

Q3 - Operations

Objective: Monitoring tech debt

Objective	Measure of success	Status
Strengthen: Improve access to and control over incident and metrics monitoring infrastructure *Team members involved: 2*	Evaluate alternative web interfaces. ~~Migrate to a web interface with improved access control over service checks and their state.~~ ~~Implement email & paging alerts for all service "owners" outside TechOps.~~ Consolidate Graphite metrics monitoring frontends into Grafana.	Goal was redefined early in the quarter. An evaluation of alternative web interfaces has been prepared but discussion & a decision hasn't completed yet. Alternative Graphite metrics monitoring frontends have been migrated into Grafana.

Learning: After failing to meet the related monitoring goal (a dependency) in the preceding quarter, we should have anticipated and adjusted this goal prior to the start of the quarter. We'll continue to work on this project in slack time (outside goals) until we are able to hire more support.

Q3 - Operations

Objective: Caching tech debt

Objective	Measure of success	Status
Experiment: Evaluate and test limited deployment of Varnish 4 *Team members involved: 2 (+1)*	Evaluate the basics of Varnish 3.x to 4.x migration for existing functionality Convert at least one cache cluster to Varnish 4.x	Wikimedia custom Varnish VCL, VMODs and related monitoring and logging tools have been ported and tested for Varnish 4. The Maps cluster has been successfully migrated to Varnish 4.

Learning: The custom *varnishkafka* integration with our logging & analytics infrastructure proved to be the biggest hurdle.

Newly hired Emanuele Rocca got up to speed very quickly and completed most of the work for this goal, with help from Luca Toscano (Analytics) for the logging dependencies.

Q3 - Operations

Other successes and misses ◐

Upgraded OTRS to version 5 and dealt with followup

Deployed minor services in codfw

Setup JupyterHub for people to access interactive notebook + terminal via the web

Migrated Ganglia to systemd

Expanded RESTbase clusters with refreshed machines, additional machines and more storage [with Services]

Assisted with getting scap3 production ready [with RelEng]

Created and documented a process for changing global traffic routing

Deployed Ferm firewalling for many additional services

Decommissioned our NetApp storage cluster

Standardized all Varnish clusters on 2-layer/multi-tier configuration

Prepared upgrade to Kubernetes 1.2 in ToolLabs

Packaged and upgraded HHVM to version 3.12 LTS [with Performance]

Prepared ORES deployment in Beta and Production [with Research]

Started maintenance of internal Linux 4.4 kernel packages and migration of jessie systems to it

Consolidated cache_mobile into cache_text

Prepared database clusters for TLS encryption for all replication

Got url-downloader into a Highly Available setup using both non-caching data centers

Large cleanup of @wikimedia.org mail aliases [with OIT]

Stabilized Labs NFS growth with per-instance resource limits

Upgraded and stabilized Salt with preparations for multi-master

Q3 - Operations

Other successes and misses

Decommissioned cache_parsoid

Refreshed codfw External Storage cluster with new hardware and more space

Refactored Varnish VCL files to reduce complexity and improve readability

Tested new Core DB hardware for upcoming eqiad refresh

Deployed Change Propagation service [with Services]

Upgraded & migrated services first to Nodejs 4.2 and then to nodejs 4.3 [with Services]

Labs DB HA setup and general maintenance

Deployed PyBal with etcd support

Migrated various services (e.g. cxserver, citoid, mathoid etc) from SCA to SCB (jessie+systemd)

Prepared 2-factor Yubikey auth with personalized keys

Hired temporary contractor to assist with DBA backlog

Promoted Chase Pettet to Engineering Manager - Labs

Q2 - Operations

Core workflows and metrics

External monitoring **availability** metrics based on Catchpoint data.

Services audience	**Availability** cumulative average over 1 day interval	
	Oct 1 - Dec 31 2015	Jan 1 - March 31 2016
(R)eaders	99.979%	**99.962%**
(C)ontent Contributors	99.974%	**99.933%**
(M)ovement Partners (volunteers, developers, etc.)	99.972%	**99.955%**
(E)xternal Partners	99.966%	**99.931%**
(D)onors	99.992%	**99.977%**

Quarterly review

Release Engineering

Q3 - 2015/16

Approximate team size during this quarter: 6

Time spent: strengthen 8%, focus 21%, experiment 0%, maintenance 71%

Key performance indicator

CI Wait Time (aka “Time to merge in MW Core”)	13.4 minutes (avg)	**-9.4% from Q2**	**-10% of last year average**

Q3 - Release Engineering

Objective: Consolidate deploy tools

Objective	**Measure of success**	**Status**
Consolidate deploy tools *Team members involved: 4*	Migrate MediaWiki to scap3 - task 114313	Not complete.

Our time in this project this quarter focused on supporting the migration of services (not MediaWiki).

With respect to our previous quarter's goal, we had to delay migration of some services to complete some feature requirements (for example: large binary support). We now have full feature parity with Trebuchet (with `git-fat` to ease transition) and have begun migrating services to scap.

Q3 - RelEng

Objective: Retire Gerrit in favor of Phabricator

Objective	Measure of success	Status
Retire Gerrit in favor of Phabricator *Team members involved: 3*	- Integrate Differential with our Continuous Integration infrastructure - task T31 - Shepherd the RFC - task T119908 - Discuss at WikiDev16 - task T114320 - **Stretch:** Garner early adopter projects (goal: 1 project per WMF "team") - task T130418	Not complete. - Integrate with CI: **NOT DONE**, goal for next quarter - Shepherd RFC: **DONE** for this quarter, though the RFC is not "approved" yet - WikiDev16: **DONE** with great success - Early adopters: 3 repositories (including scap)

2 of our 3 quarterly sub-goals were met. The one which was not was due to, simply, a lack of time to complete the work given other priorities (the next goal, reducing CI wait time).

Special callout to the great success of the Differential RFC discussion at WikiDev16 in January.

Q3 - Release Engineering

Objective: Reduce CI wait time

Objective	**Measure of success**	**Status**
Reduce CI wait time *Team members involved: 1*	Migrate remaining CI jobs to Nodepool - task T119138 ● php composer (Zend and HHVM) - task T119139 ● as many miscellaneous jobs as possible - task T119140 ● javascript npm jobs - task T119143	Not complete. ● Composer: **NOT DONE** ● Misc jobs: **NOT DONE** ● Javascript npm: **DONE** except for: ○ pywikibot i18n ○ Parsoid ○ OOJS UI

The vast majority of Javascript/npm CI jobs have migrated to our nodepool infrastructure, however, the php/composer CI jobs are blocked on an infrastructure need (HHVM on Jessie, or Trusty nodepool images).

Thanks to the volunteer Paladox for his help in migrating a large number of the npm jobs himself.

Q3 - Release Engineering

Other successes and misses

Successes:

- Cleaned up (removed or fixed) a lot of browser tests that were languishing. - T94150
- Almost finished with making it easier to define daily browser test runs - T128190
- Upgraded Jenkins from 1.625.3 to LTS 1.642.2 - T128058
- Added git-fat support in scap3 - T129420
- Switched Phabricator to deploy with scap3 - T114363
- Switched AQS to deploy with scap3 - T114999
- Added 3 new members to the SWAT team: Dereckson, Katie (aude), MaxSem
    - (2 others in-training)
- Extra on-site time post-All Hands in SF

Misses:

- See previous slides :)

Q3 - Release Engineering

Core workflows and metrics

Category	Workflow	Comments	Type
Service maintenance	Phabricator maintenance	Phabricator upgrades (bi-weekly, as needed), and user account help	M
	MediaWiki + SWAT deployments	Daily SWAT deployments Weekly ("train") production deploys of MW+Extensions MW train deploys now a rolling responsibility within the team	M

Type: new, reactive, maintenance

Q3 - Release Engineering

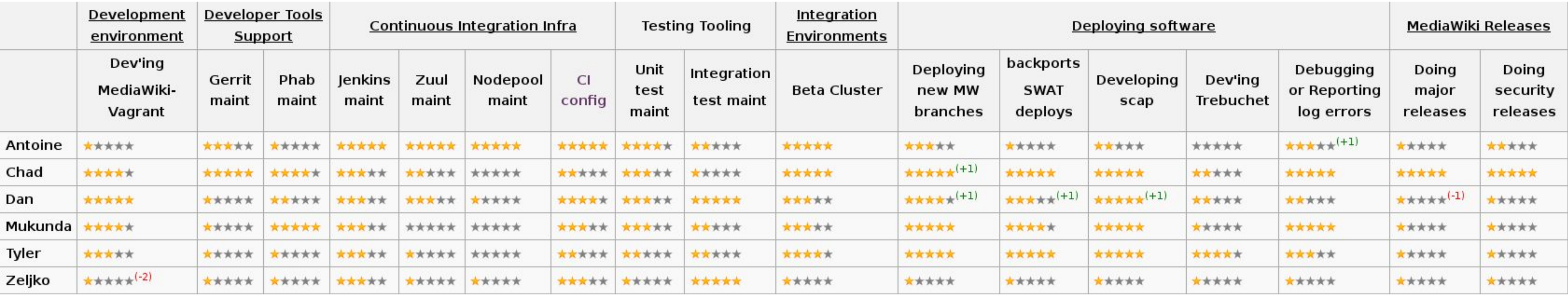
Core workflows and metrics

Category	Workflow	Comments	Type
Continuous Integration / Test maintenance	CI Config changes	Changes to our CI configuration to enable/disable tests in a repository, etc ● Jan: 12 opened, 14 closed (-4) ● Feb: 36 opened, 30 closed (+6) ● Mar: 19 opened, 15 closed (+4)	R
	MediaWiki Selenium (automated browser tests library)	2 releases in Q3 ● https://rubygems.org/gems/mediawiki_selenium/versions	M
	Browser Test maintenance	Removing unused tests T94150, updating libraries, etc	M

Type: new, reactive, maintenance

Q3 - Release Engineering

SPOF tracking : Skill Matrix



	Development environment	Developer Tools Support		Continuous Integration Infra				Testing Tooling		Integration Environments	Deploying software					MediaWiki Releases	
	Dev'ing MediaWiki-Vagrant	Gerrit maint	Phab maint	Jenkins maint	Zuul maint	Nodepool maint	CI config	Unit test maint	Integration test maint	Beta Cluster	Deploying new MW branches	backports SWAT deploys	Developing scap	Dev'ing Trebuchet	Debugging or Reporting log errors	Doing major releases	Doing security releases
Antoine	★☆☆☆☆	★★★☆☆	★☆☆☆☆	★★★★★	★★★★★	★★★★★	★★★★★	★★★★☆	★★☆☆☆	★★★★★	★★★☆☆	★☆☆☆☆	★★☆☆☆	☆☆☆☆☆	★★★☆☆(+1)	★☆☆☆☆	★★☆☆☆
Chad	★★★★☆	★★★★★	★★★★☆	★★★☆☆	★★☆☆☆	☆☆☆☆☆	★★☆☆☆	★★★☆☆	★☆☆☆☆	★★★★★	★★★★★(+1)	★★★★★	★★★★★	★★☆☆☆	★★★★★	★★★★★	★★★★★
Dan	★★★★★	★☆☆☆☆	★★☆☆☆	★★★☆☆	★★★☆☆	★☆☆☆☆	★★★★☆	★★★☆☆	★★★★★	★★★☆☆	★★★★☆(+1)	★★★☆☆(+1)	★★★★★(+1)	★★☆☆☆	★★☆☆☆	★☆☆☆☆(-1)	★☆☆☆☆
Mukunda	★★★★☆	★☆☆☆☆	★★★★★	★★★☆☆	☆☆☆☆☆	☆☆☆☆☆	★★★☆☆	★★★☆☆	★★☆☆☆	★★★☆☆	★★★★★	★★★★☆	★★★★★	★☆☆☆☆	★★★★★	★☆☆☆☆	★☆☆☆☆
Tyler	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★★☆☆	★☆☆☆☆	☆☆☆☆☆	★★☆☆☆	★★☆☆☆	★★☆☆☆	★★★★☆	★★★★★	★★★★★	★★★★★	★★★★☆	★★★☆☆	★☆☆☆☆	★☆☆☆☆
Zeljko	★☆☆☆☆(-2)	★☆☆☆☆	★☆☆☆☆	★★★☆☆	★☆☆☆☆	★☆☆☆☆	★★★☆☆	★☆☆☆☆	★★★★★	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆	★☆☆☆☆

Warning: Self-reported data.

Quarterly review

Security

Q3 - 2015/16

Approximate team size during this quarter: 2.25

Time spent: core 90%, strengthen 10%

Key performance indicator

Critical and High Priority Security Bugs	(redacted)	-50% / -3.6% from Q2	n/a

Q3 - Security

Objective: 2FA for CentralAuth wikis

Objective	Measure of success	Status
Improve authentication on WMF wikis by allowing users to enable two-factor authentication for their accounts (T107605)	CentralAuth accounts will have the option to enable two-factor authentication.	Incomplete by April 1st

Learning: Under staffing did not allow the team to absorb an influx of new security issues on our sites, delaying implementation of this feature.

Q3 - Security

Other successes and misses

Success: Passed Password Policy RFC on meta, strengthening password policy for administrators, bureaucrats, checkusers, and oversighters.

Miss: Not able to complete one of the committed security reviews

Q3 - Security

Core workflows and metrics

Category	Workflow	Comments	Type
MediaWiki Security	Security Bugs	(redacted) (+76% from Q2) valid issues opened, including (redacted) (+61% from Q2) UBN/High issues.	R
	Security Reviews	19 opened, performed 10	R
	Security Scanning	Weekly static and dynamic security scanning	M
Privacy and Security Consulting	Architecture and design consultations	* Involvement in the Architecture Summit * Continued work with Reading Infrastructure on Auth/Session Manager * Continued work with Privacy on Data Access Policies, Privacy by Design * Implemented 2FA for Horizon with Ops	N

Type: new, reactive, maintenance

Q3 - Security

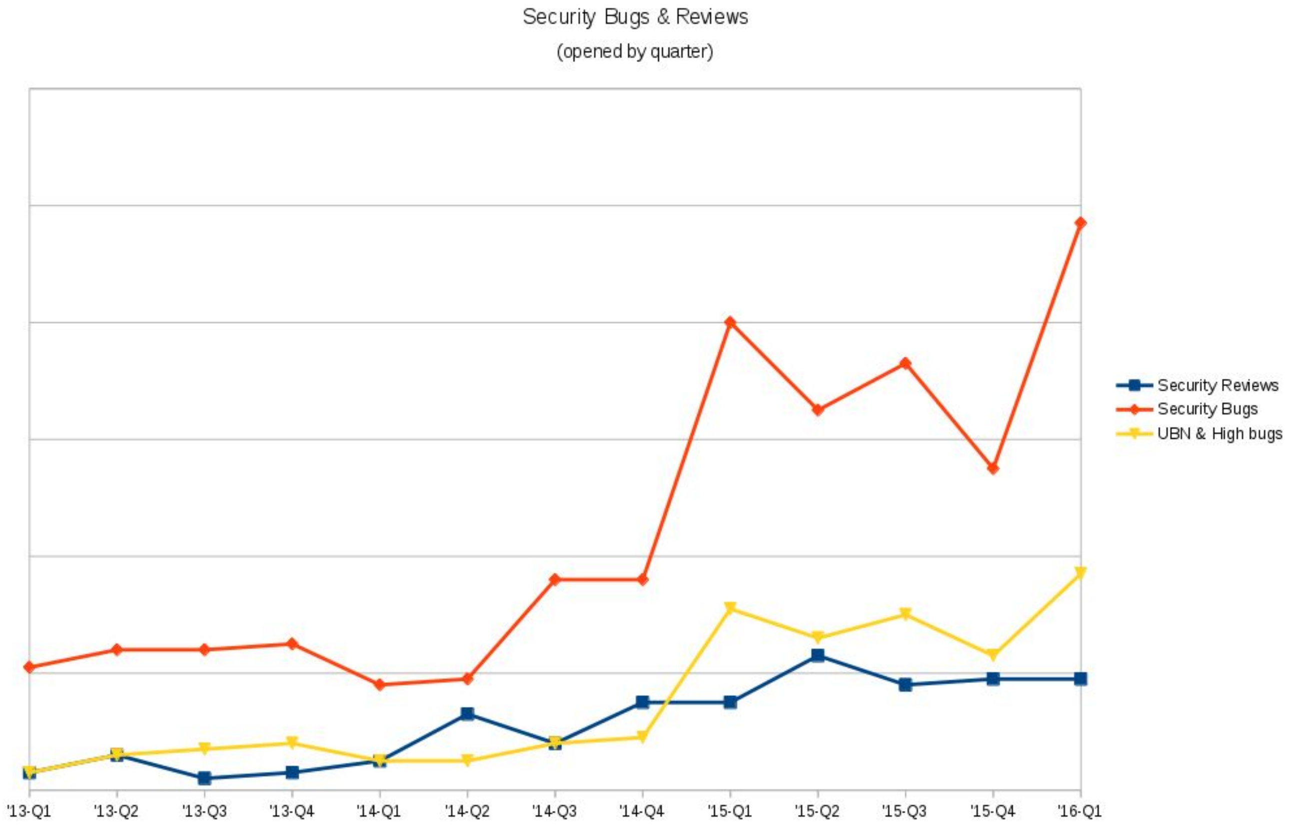
Core workflows and metrics

Category	Workflow	Comments	Type
DFIR	Incident Response	(redacted)	R
Training	Secure Coding	PCI: 5 Attendees	M
	Privacy, Security, Licensing, & Beyond	63 viewers	N
Compliance	Monthly vulnerability scan for PCI	OpenVAS scans of frack were run monthly	M

Type: new, reactive, maintenance

Q3 - Security

Bugs opened / quarter



Quarterly review

Services

Q3 - 2015/16

Approximate team size during this quarter: 4 FTE

Time spent: strengthen 50%, focus 35%, experiment 15%

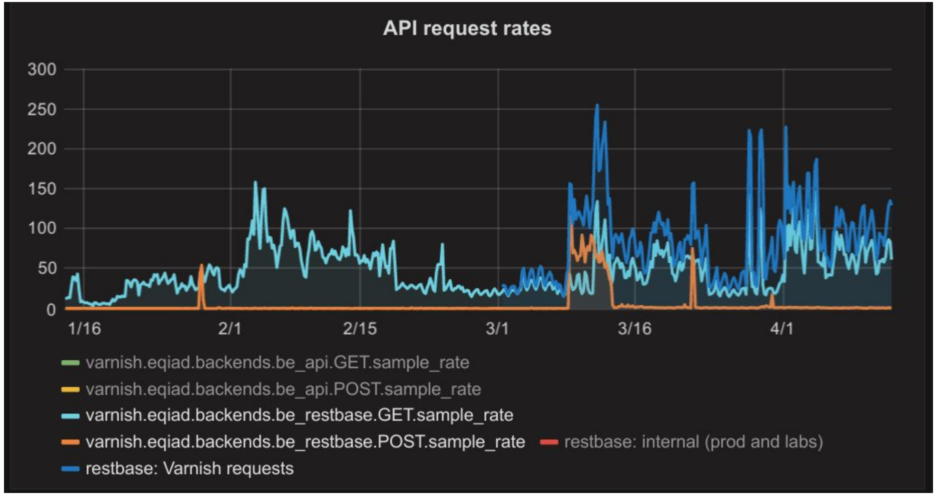
Key performance indicators

REST API requests (all)	440 req/s mean	+77% up from 248 req/s Q2	
REST API requests, without update requests	323 req/s mean ~84 external, ~230 req/s prod + labs	No data for Q2	
REST API uptime	99.961%	-0.009% from Q2	

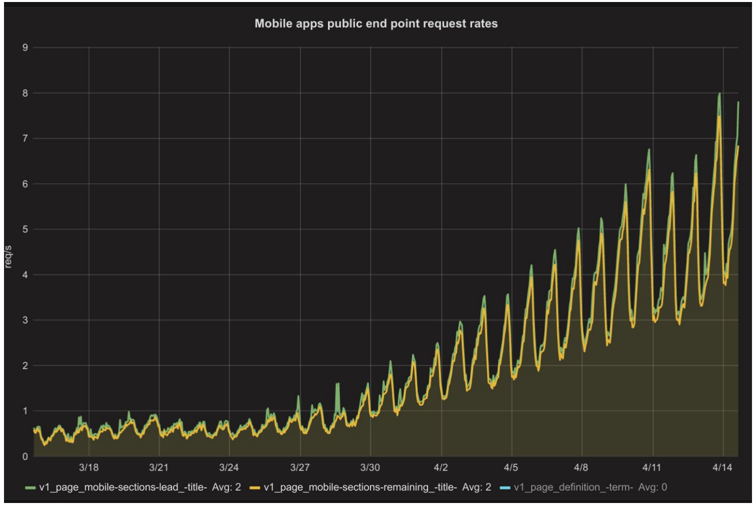
Q3 - Services

Appendix: REST API request rates

REST API request rates: Varnish



REST API request rates: Mobile section requests (Android app roll-out)



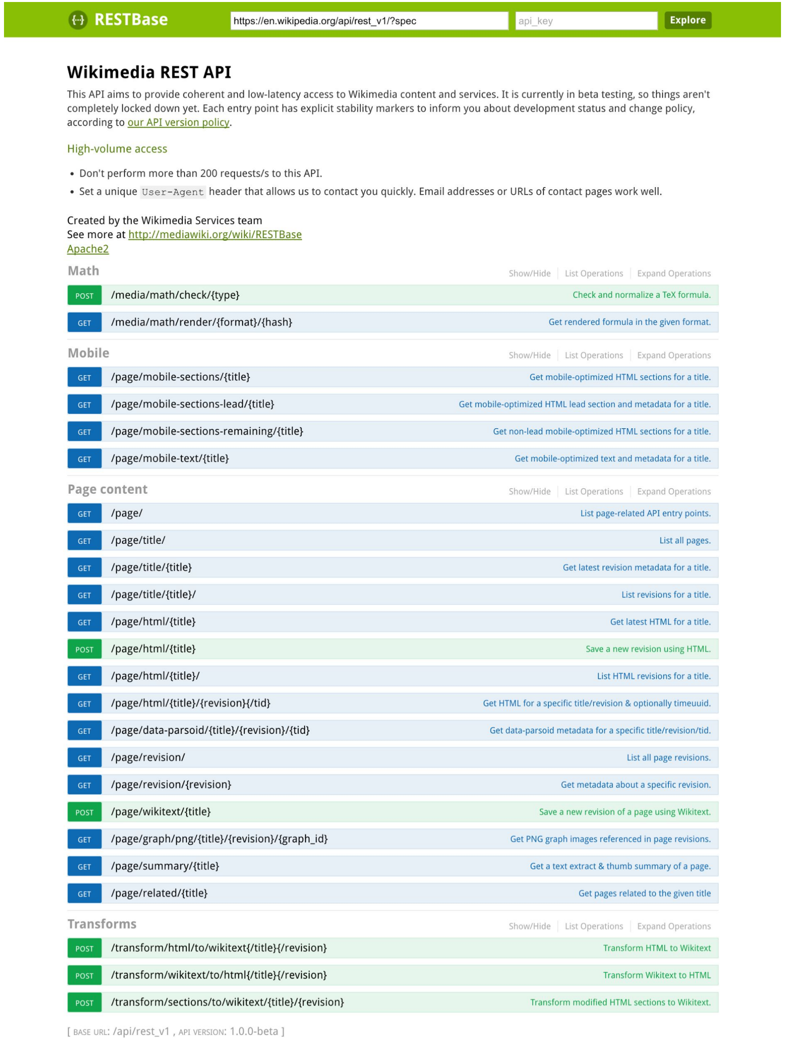
Q3 - Services

Objective: API & community build-out

Objective	Measure of success	Status
Strengthen: **REST API expansion & documentation.** *Team members involved: 4*	Refined guidelines and documentation to guide new service developers with less one-off mentoring.	Expanded docs to cover development to deployment. Created documentation index page. Documented API policies.
	Support apps and the API-driven frontend effort with cacheable high-traffic entry points and the definition of content composition interfaces.	Integrated RESTBase and change propagation with CDN purging. Android app powered by cached & purged REST API end points backed by Mobile Content Service (Reading). Created RFCs & code for section wrapping, component matching.

Endpoints like "page summary" or "wiktionary definition" are (on purpose) very general, and are picked up by other internal & external consumers (ex: hovercards).

Learning: We need a thumb API that lets *clients* select thumb sizes to make end points like page summary more flexible. Wrote RFC, in discussion.



RESTBase   https://en.wikipedia.org/api/rest_v1/?spec   api_key   Explore

Wikimedia REST API

This API aims to provide coherent and low-latency access to Wikimedia content and services. It is currently in beta testing, so things aren't completely locked down yet. Each entry point has explicit stability markers to inform you about development status and change policy, according to our API version policy.

High-volume access

- Don't perform more than 200 requests/s to this API.
- Set a unique `User-Agent` header that allows us to contact you quickly. Email addresses or URLs of contact pages work well.

Created by the Wikimedia Services team
See more at http://mediawiki.org/wiki/RESTBase
Apache2

Math

Show/Hide | List Operations | Expand Operations

POST	/media/math/check/{type}	Check and normalize a TeX formula.
GET	/media/math/render/{format}/{hash}	Get rendered formula in the given format.

Mobile

Show/Hide | List Operations | Expand Operations

GET	/page/mobile-sections/{title}	Get mobile-optimized HTML sections for a title.
GET	/page/mobile-sections-lead/{title}	Get mobile-optimized HTML lead section and metadata for a title.
GET	/page/mobile-sections-remaining/{title}	Get non-lead mobile-optimized HTML sections for a title.
GET	/page/mobile-text/{title}	Get mobile-optimized text and metadata for a title.

Page content

Show/Hide | List Operations | Expand Operations

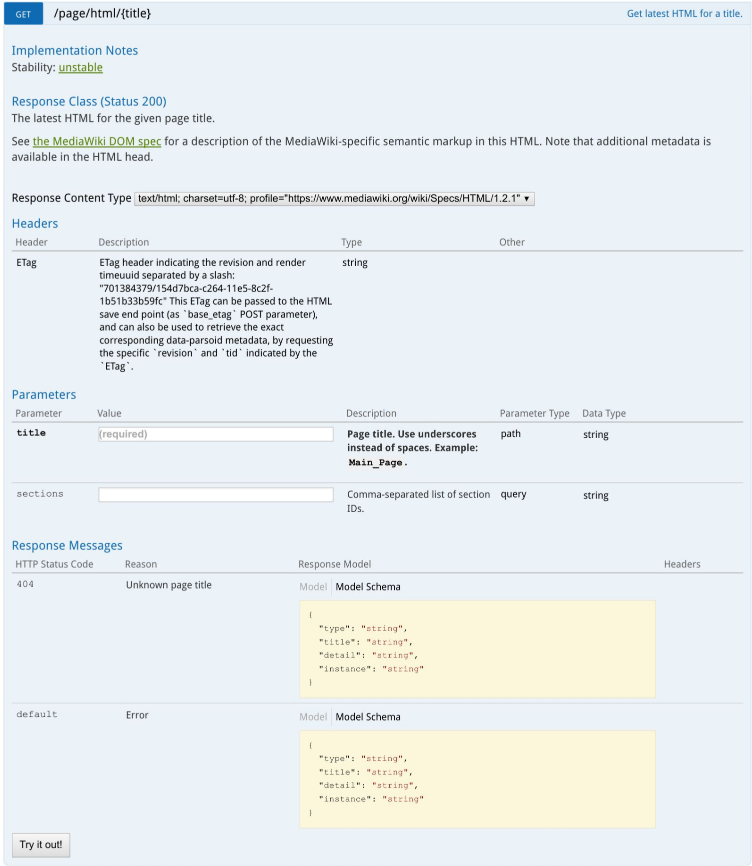
GET	/page/	List page-related API entry points.
GET	/page/title/	List all pages.
GET	/page/title/{title}	Get latest revision metadata for a title.
GET	/page/title/{title}/	List revisions for a title.
GET	/page/html/{title}	Get latest HTML for a title.
POST	/page/html/{title}	Save a new revision using HTML.
GET	/page/html/{title}/	List HTML revisions for a title.
GET	/page/html/{title}/{revision}{/tid}	Get HTML for a specific title/revision & optionally timeuuid.
GET	/page/data-parsoid/{title}/{revision}/{tid}	Get data-parsoid metadata for a specific title/revision/tid.
GET	/page/revision/	List all page revisions.
GET	/page/revision/{revision}	Get metadata about a specific revision.
POST	/page/wikitext/{title}	Save a new revision of a page using Wikitext.
GET	/page/graph/png/{title}/{revision}/{graph_id}	Get PNG graph images referenced in page revisions.
GET	/page/summary/{title}	Get a text extract & thumb summary of a page.
GET	/page/related/{title}	Get pages related to the given title

Transforms

Show/Hide | List Operations | Expand Operations

POST	/transform/html/to/wikitext{/title}{/revision}	Transform HTML to Wikitext
POST	/transform/wikitext/to/html{/title}{/revision}	Transform Wikitext to HTML
POST	/transform/sections/to/wikitext/{title}/{revision}	Transform modified HTML sections to Wikitext.

[BASE URL: /api/rest_v1 , API VERSION: 1.0.0-beta]



GET /page/html/{title}   Get latest HTML for a title.

Implementation Notes

Stability: unstable

Response Class (Status 200)

The latest HTML for the given page title.

See the MediaWiki DOM spec for a description of the MediaWiki-specific semantic markup in this HTML. Note that additional metadata is available in the HTML head.

Response Content Type text/html; charset=utf-8; profile="https://www.mediawiki.org/wiki/Specs/HTML/1.2.1" ▾

Headers

Header	Description	Type	Other
ETag	ETag header indicating the revision and render timeuuid separated by a slash: "701384379/154d7bca-c264-11e5-8c2f-1b51b33b59fc" This ETag can be passed to the HTML save end point (as `base_etag` POST parameter), and can also be used to retrieve the exact corresponding data-parsoid metadata, by requesting the specific `revision` and `tid` indicated by the `ETag`.	string	

Parameters

Parameter	Value	Description	Parameter Type	Data Type
title	(required)	**Page title. Use underscores instead of spaces. Example: `Main_Page`.**	path	string
sections		Comma-separated list of section IDs.	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
404	Unknown page title	Model | Model Schema { "type": "string", "title": "string", "detail": "string", "instance": "string" }	
default	Error	Model | Model Schema { "type": "string", "title": "string", "detail": "string", "instance": "string" }	

Try it out!

Q3 - Services

Objective: Storage scaling, selective replication for hot data

Objective	Measure of success	Status
Focus: **Storage scaling & selective replication for hot data.** *Team members involved: 4*	Reduce cost of history storage by improving compression ratios for HTML content to < 10% of raw content size.	Improved HTML revision compression from **~17% to ~3%** using Brotli algorithm, reducing storage needs to **less than ⅕.**
	Support selective replication of hot data to edge PoPs for low-latency API access.	Separated current version storage from old revisions. Sped up requests for current revisions by 50+% by avoiding gzip recompression.

Deploys to production blocking on Cassandra cluster expansion (ETA: ~1 month). Pace of cluster expansion partly determined by ops availability.

Learnings:

Alternative strategy of chunked storage introduces overly large read performance penalty at chunk sizes < 1mb.

Brotli looks like an all-round win for both CPU time (compression / decompression) and storage size. Only downside: Memory usage with large compression blocks.

Q3 - Services

Objective: Reliable event production & change propagation.

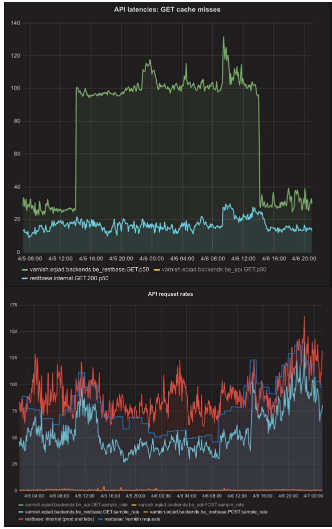
Objective	Measure of success	Status
Experiment: **EventBus & change propagation.** *Team members involved: 2*	Wrap up EventBus. Deploy change propagation service.	MediaWiki edit events & RESTBase update events produced to EventBus. EventBus prepared for multi-DC operation. Change propagation service deployed end of quarter. Resource change topic introduced.
	Implement ~~reliable~~ event production from MediaWiki to EventBus. Migrate RESTBase update jobs & cache updates to EventBus consumers. Provide a lightweight EventBus implementation for third party users. Prototype fine-grained dependency tracking for change propagation.	Not done.

Other tasks (DC fail-over, service deployments) took priority.

Q3 - Services

[+] Multi-DC support: Support for multiple DCs (including active/active) has been a major design consideration for RESTBase and stateless services from the beginning. As a result, we were able to failover DCs for services (RESTBase, Parsoid, mobile content service, mathoid, graphoid) twice without user impact.

[+] Reliable deploys and testing: High test coverage (93%) and careful deploy procedures (using beta labs, staging cluster and canaries) resulted in no VisualEditor corruptions, and generally outage-free deploys despite significant clean-up refactoring. The exception was a temporary end point breakage affecting the pageview API. This was quickly fixed, and the endpoint was covered by tests.



[+] API result format versioning: Through an RFC, we decided on a format versioning strategy for the REST API. This is now documented, and will enable an upcoming major HTML format change without breaking existing clients.

Q3 - Services

Core workflows and metrics

Category	Workflow	Comments	Type
Service development & operations	Guidance and code review for existing services	- **MobileApps**: Helped the Reading team expand mobileapps / content service functionality, and track . - **Math**: Worked with volunteer Moritz Schubotz and MathJax community towards a crisper and more accessible Math render mode using SVG and MathML. German Wikibooks are now asking for this to be made the default render mode, and work to make this happen is underway. - **Service ops**: Shepherded service deployments, investigated issues & improved shared operational service infrastructure. - **AQS**: Worked with Analytics team to investigate performance issues in pageview API (unique devices), expand functionality. - **Citoid**: Mentored Citoid development (Marielle Volz). Parsing team took over maintenance by end of quarter. Approx effort: 0.9 FTE	R/M
	Cassandra operations	Worked with operations to expand Cassandra capacity, upgrade to new versions & refine the configuration. Help other Cassandra users like Analytics in diagnosing & addressing issues. Approx effort: 1 FTE	M

Type: new, reactive, maintenance

Appendix

Q3 - Architecture

Key questions from WikiDev16 (slide 1 of 2)

The “5 key questions” we defined for WikiDev16 based on the RFCs we had at the time were useful for organizing the summit, but haven’t served as a lasting organizational tool:

1. **Content format** (T119022) - This is about the format of the authoritative format for the source data of our site (e.g. wikitext, JSON, and accompanying database fields). The central problem in this area: "how do we make manipulating our data easier and more useful" (both for humans and computers)

2. **Content access and APIs** (T119029) - this is about getting our data in-and-out of the system (e.g. rest.wikimedia.org). The central problem in this area: "how do we make accessing and distributing our data easier and more useful?"

3. **Collaboration** (T119030) - this is about how we work together. Central problem: "how do we scale editing our code up to populations similar to editing our projects, proportionally increasing our positive impact and productivity?"

Q3 - Architecture

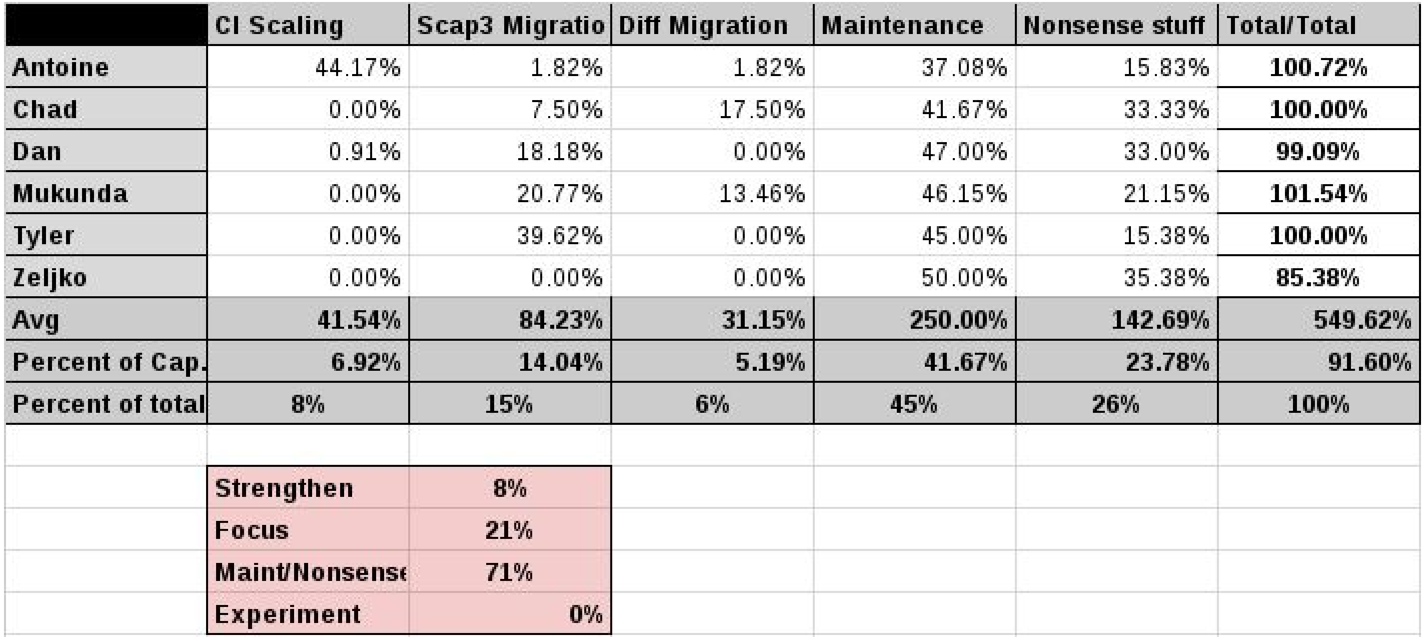
Key questions from WikiDev16 (slide 2 of 2)

4. **Software engineering** (T119032) - this is about building and delivering high quality code. Central problem: "how do we build high-quality software that we can dramatically increase the number of people that can understand it while increasing the reliability and maintainability of Wikimedia sites?"

5. **User interface presentation** (T119162) - improving our user interactions. Central problem: "how to we make our software look and feel joyful to use?"

Q3 - Release Engineering

Appendix : Time spent



	CI Scaling	Scap3 Migratio	Diff Migration	Maintenance	Nonsense stuff	Total/Total
Antoine	44.17%	1.82%	1.82%	37.08%	15.83%	**100.72%**
Chad	0.00%	7.50%	17.50%	41.67%	33.33%	**100.00%**
Dan	0.91%	18.18%	0.00%	47.00%	33.00%	**99.09%**
Mukunda	0.00%	20.77%	13.46%	46.15%	21.15%	**101.54%**
Tyler	0.00%	39.62%	0.00%	45.00%	15.38%	**100.00%**
Zeljko	0.00%	0.00%	0.00%	50.00%	35.38%	**85.38%**
Avg	**41.54%**	**84.23%**	**31.15%**	**250.00%**	**142.69%**	**549.62%**
Percent of Cap.	**6.92%**	**14.04%**	**5.19%**	**41.67%**	**23.78%**	**91.60%**
Percent of total	**8%**	**15%**	**6%**	**45%**	**26%**	**100%**

Strengthen	8%
Focus	21%
Maint/Nonsense	71%
Experiment	0%

Warning: Self-reported data.