MEZA

Sweet!

MediaWiki E-Z Administration

Project founded by great people at NASA
Johnson Space Center (JSC)

Now led by Rich Evans
NASA
Armstrong Test Facility (ATF)
Goddard Research Center (GRC)

Greg Rundlett

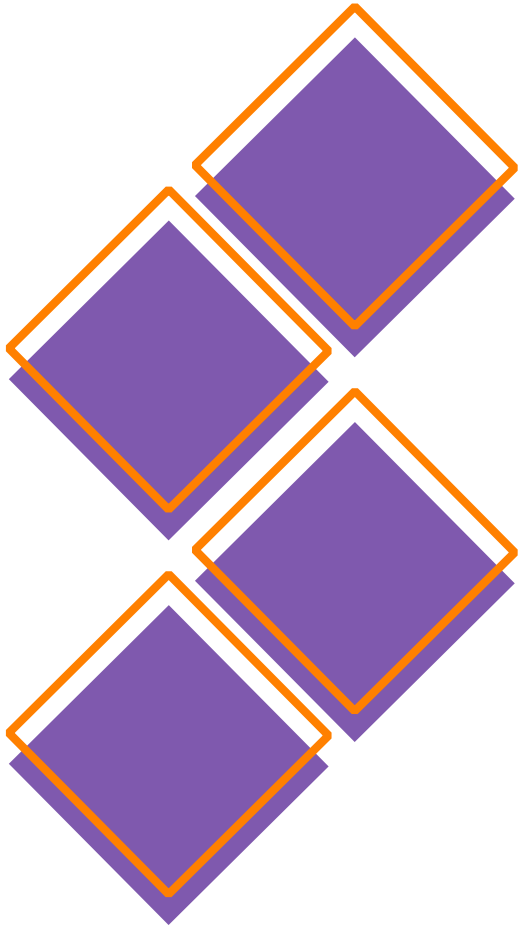
1 MEZA what?

MEZA who? 2

3 MEZA why?

Current Status 4

5 Future



1

MEZA is

an Ansible-driven, Infrastructure as Code (IaC) complete platform for Wiki farms.

Meza is a push-button solution to Knowledge Management.

The entire MediaWiki ecosystem in convenient spill-proof packaging

Work by: Falia Vicky

Language	Files	Lines	Blanks	Comments	Code	Complexity
YAML	125	9594	1272	1775	6547	0
Jinja	74	7832	1492	22	6318	280
Shell	26	1901	376	542	983	132
Alex	18	20	0	0	20	0
Plain Text	15	1366	244	0	1122	0
PHP	10	3747	636	486	2625	242
Markdown	7	942	249	0	693	0
Dockerfile	4	167	16	9	142	58
JavaScript	4	22950	1971	1307	19672	3170
SQL	3	913	114	4	795	0
JSON	2	218	0	0	218	0
Python	2	1841	26	100	1715	26
CSS	1	641	129	30	482	0
INI	1	1	0	0	1	0
License	1	22	5	0	17	0
gitignore	1	17	5	5	7	0
Total	**294**	**52172**	**6535**	**4280**	**41357**	**3908**

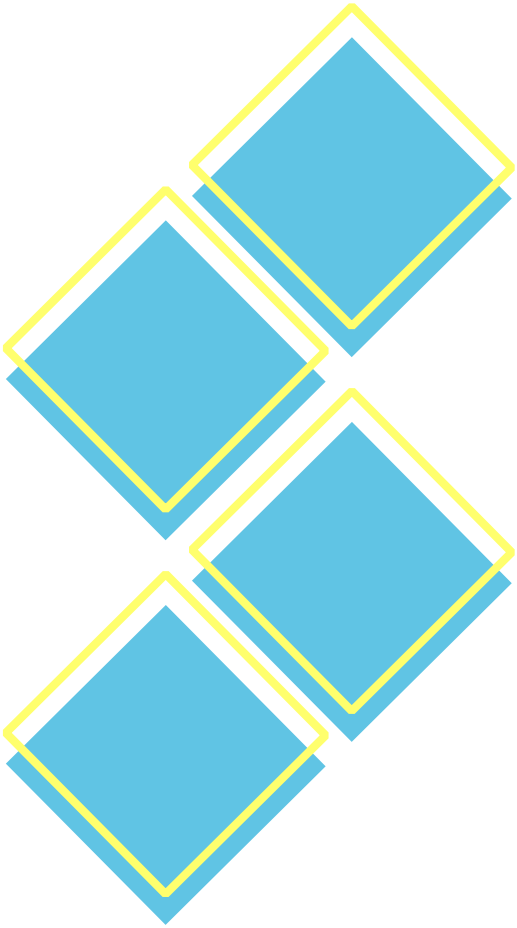
Estimated Cost to Develop (organic) **$2,783,081**
Estimated Schedule Effort (organic) 15.40 months
Estimated People Required (organic) 7.77

2

Meza Who

Whether you are a MediaWiki consultant, hoster, or SysAdmin, Meza is a healthy part of your balanced diet.

Meza is for you.



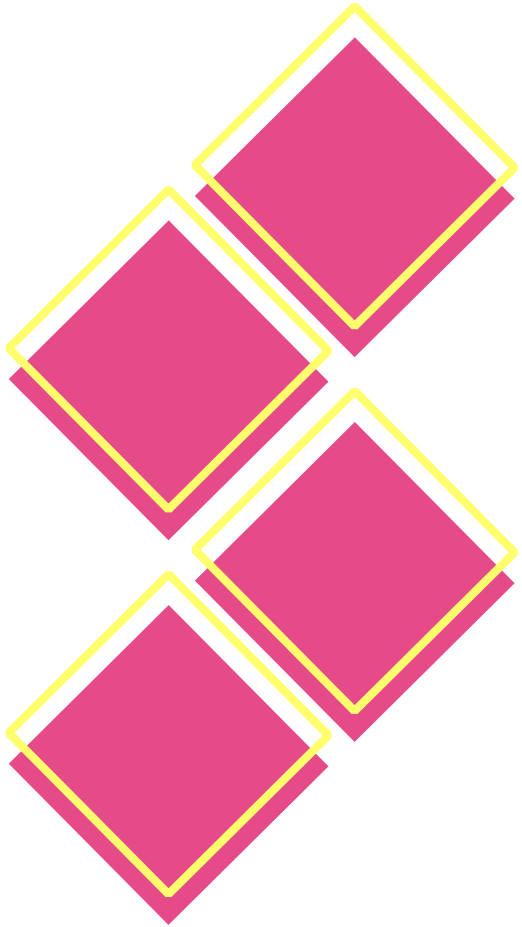
3

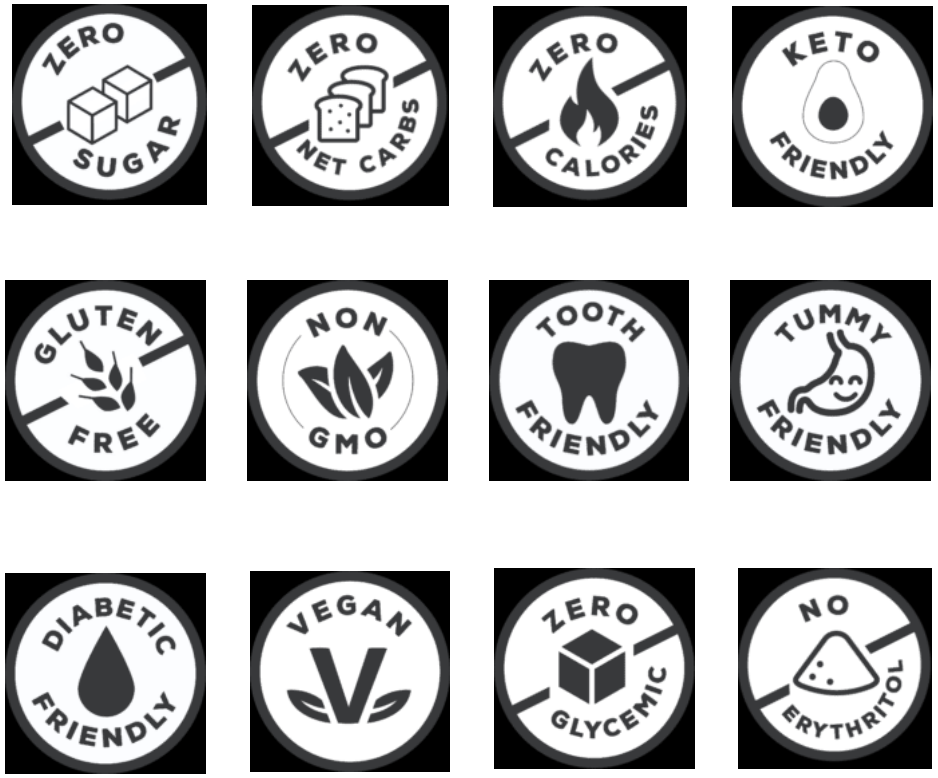
Why

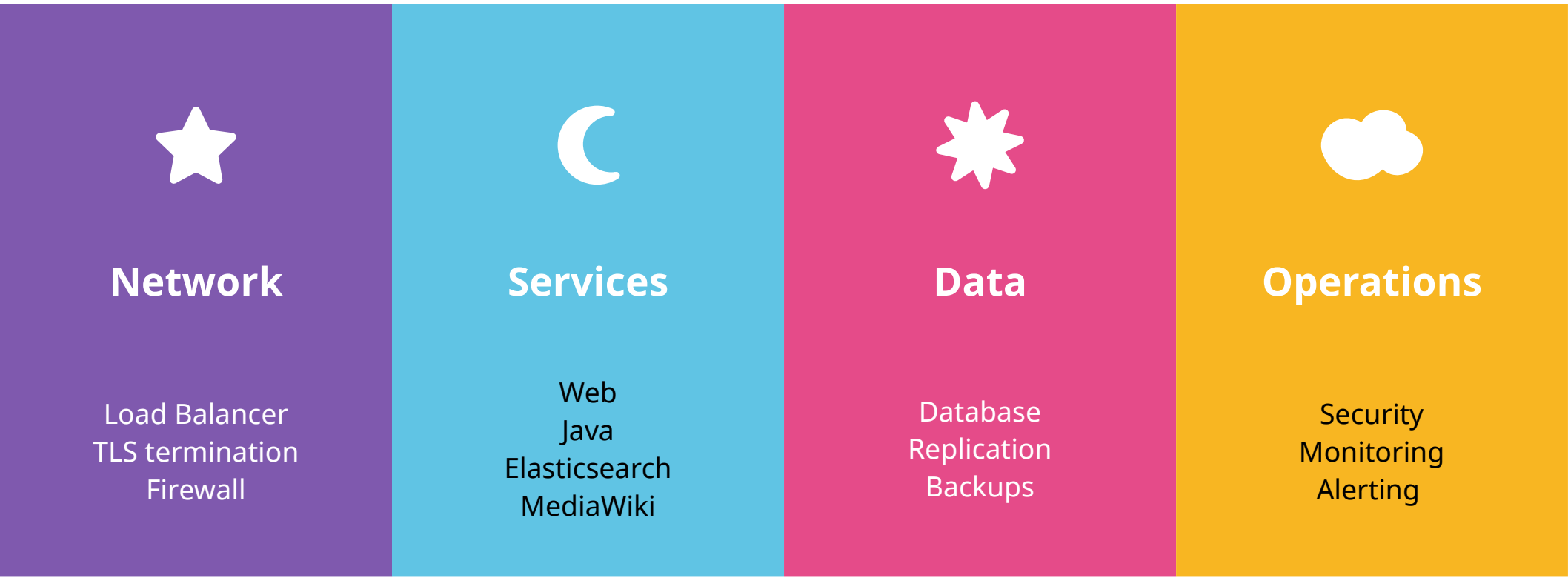
Wholesome ingredients:

Sugar free
Caffeine free
Alcohol free

And no fees!
(I’m looking at you Confluence and SharePoint)




ZERO SUGAR
ZERO NET CARBS
ZERO CALORIES
KETO FRIENDLY
GLUTEN FREE
NON GMO
TOOTH FRIENDLY
TUMMY FRIENDLY
DIABETIC FRIENDLY
VEGAN
ZERO GLYCEMIC
NO ERYTHRITOL


Network
Load Balancer
TLS termination
Firewall
Services
Web
Java
Elasticsearch
MediaWiki
Data
Database
Replication
Backups
Operations
Security
Monitoring
Alerting

Tasks / Responsibilities

- Operating System
- Apache + modules
- PHP + modules
- Composer +
- Python
- Automation
- Backups
- MediaWiki

- Code updates
- Database updates
- Configure wikis
- Create new wikis
- Delete wikis
- Task runner
- Elasticsearch
- Firewall

- Distributed filesystem
- Proxy
- Web Content
- Image processing
- Security
- Lua
- Cache

- Logging
- Monitoring
- Cluster sync
- SAML / Auth

Extensions

AdminLinks
AdvancedSearch
Arrays
Bootstrap
CharInsert
CirrusSearch
Cite
CodeEditor
CodeMirror
CollapsibleVector
CommentStreams
ContributionScores

CreateUserPage
CSS
DataTransfer
DeleteBatch
DismissableSiteNotice
DisplayTitle
DynamicSidebar
Echo
Elastica
ExternalData
FlexDiagrams
Flow

Graph
HeaderFooter
HeaderTabs
ImageMap
InputBox
Interwiki
LabeledSectionTransclusion
Lockdown
Maps
MediaFunctions
Mermaid

ModernTimeline
MultimediaViewer
MyVariables
Network
NumerAlpha
OpenLayers

(continued)

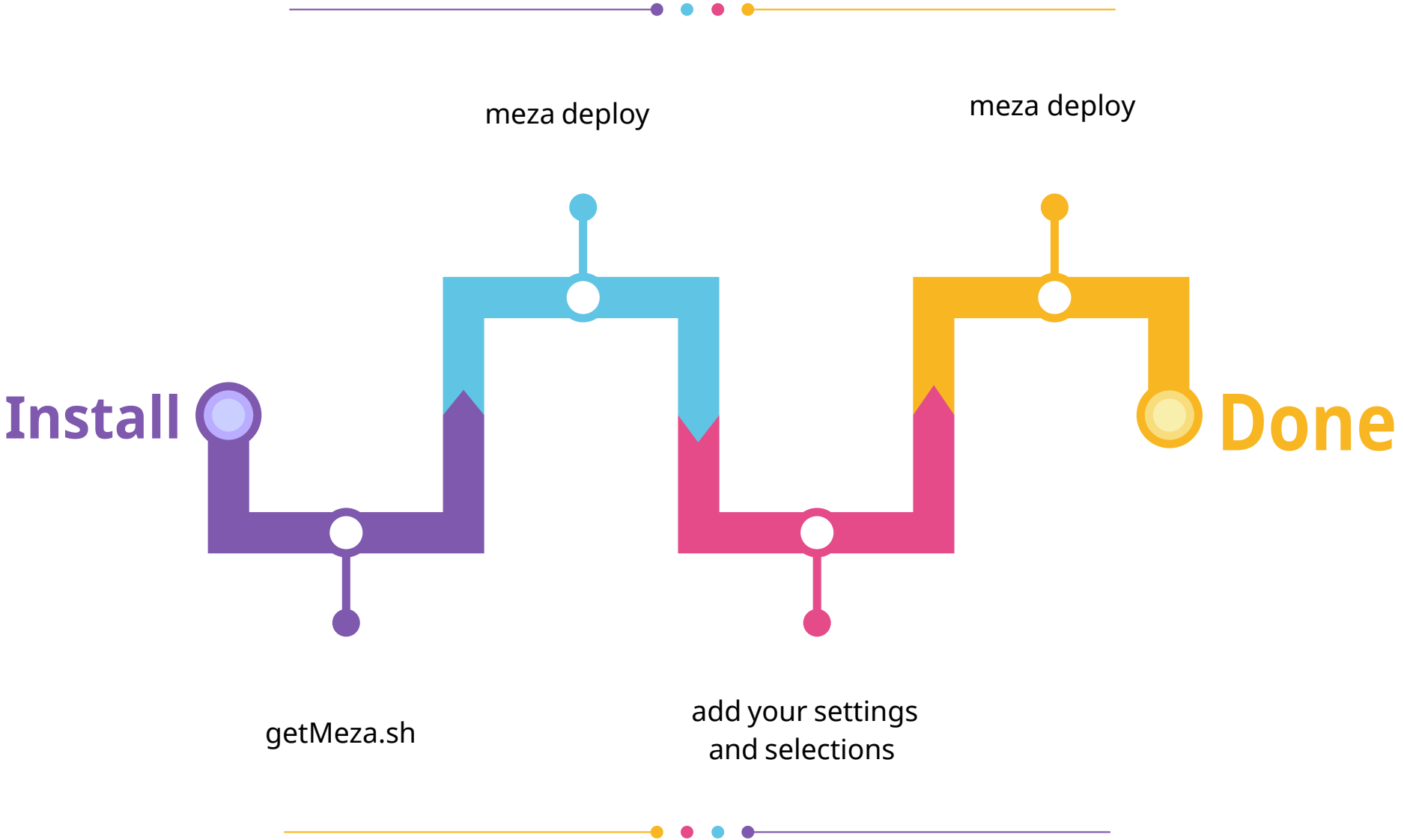
Extensions

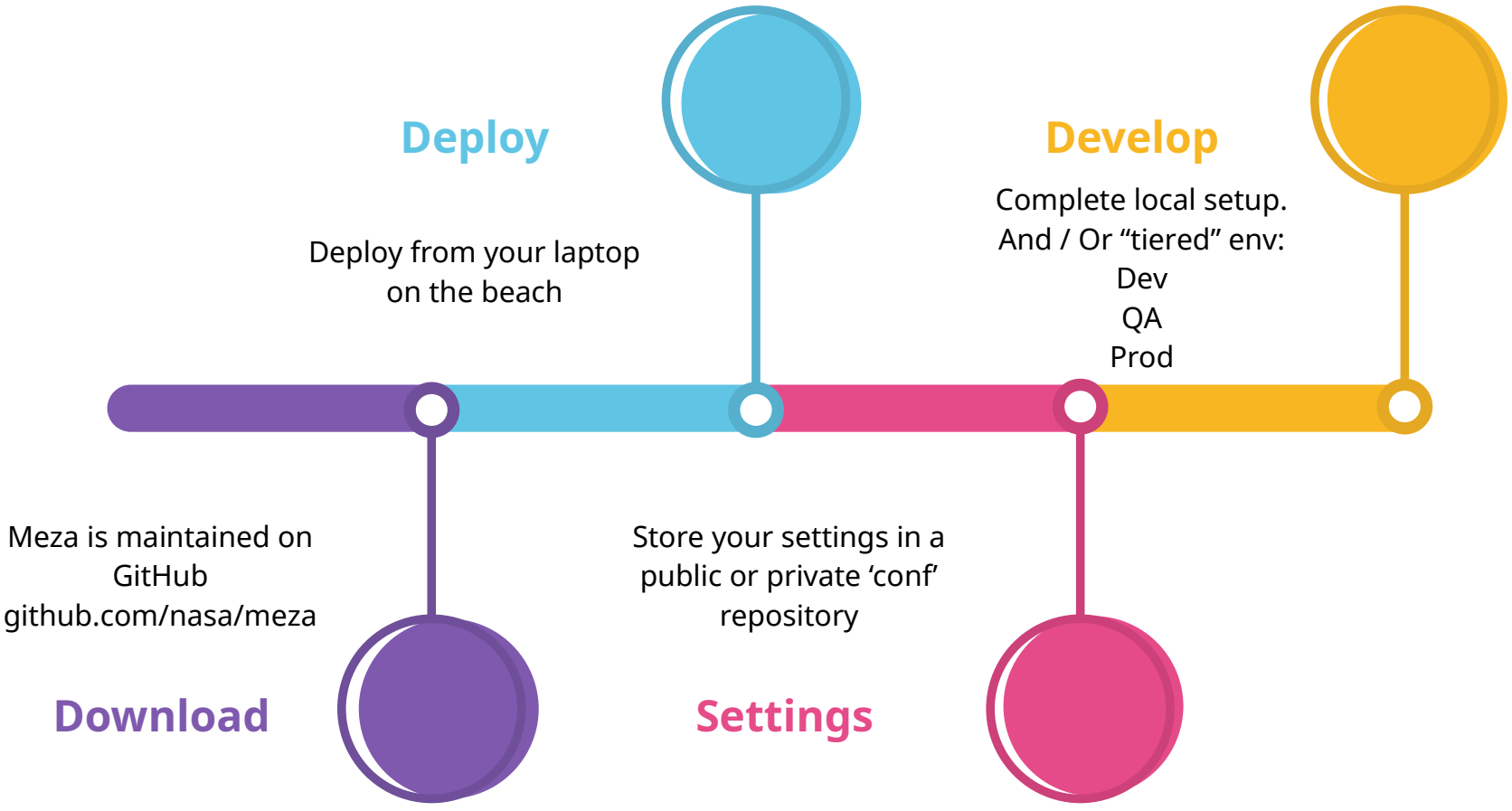
PageForms
PageImages
ParserFunctions
PipeEscape
ReplaceText
RevisionSlider
Scribunto
CompoundQueries
SemanticDrilldown

ExtraSpecialProperties
Semantic MediaWiki
Semantic Result Formats
Semantic Scribunto
SimpleBatchUpload
SimpleMathJax
SubpageFun

SubPageList
SyntaxHighlight_GeSHi
TemplateData
TextExtracts
Thanks
LanguageSelector
Variables
VEForAll

VisualEditor
WatchAnalytics
WhitelistPages
WhoIsWatching
WhosOnline
Widgets
WikiEditor
YouTube


meza deploy
meza deploy
Install
Done
getMeza.sh
add your settings
and selections


Deploy
Deploy from your laptop on the beach
Develop
Complete local setup.
And / Or “tiered” env:
Dev
QA
Prod
Meza is maintained on GitHub
github.com/nasa/meza
Download
Store your settings in a public or private ‘conf’ repository
Settings

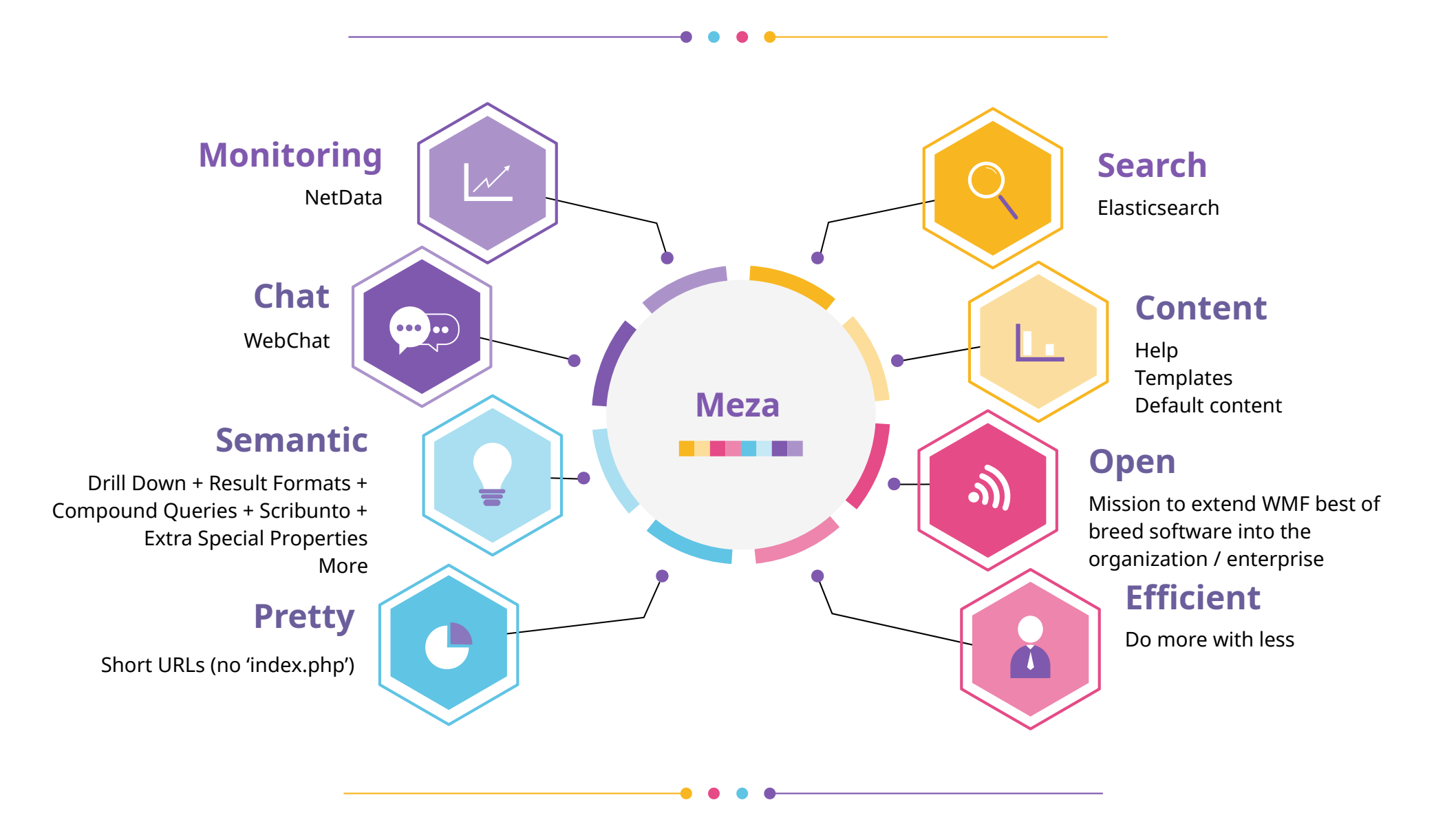
32 Flavors

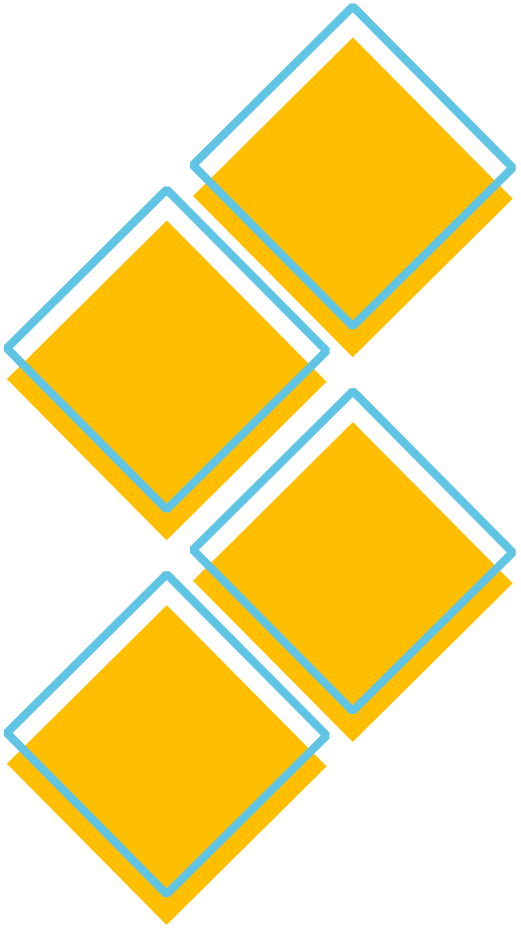
Multiple branches for platform versions and modules

X

configurations for each farm member

Work by: Muhammad Aria


Monitoring
NetData
Chat
WebChat
Semantic
Drill Down + Result Formats + Compound Queries + Scribunto + Extra Special Properties More
Pretty
Short URLs (no 'index.php')
Meza
Search
Elasticsearch
Content
Help
Templates
Default content
Open
Mission to extend WMF best of breed software into the organization / enterprise
Efficient
Do more with less

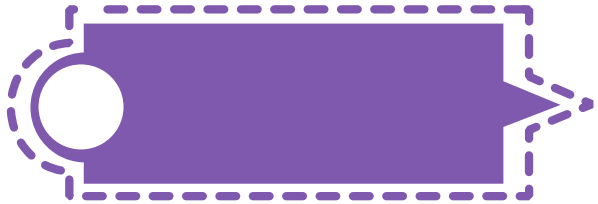


Status

Hot, out of the oven

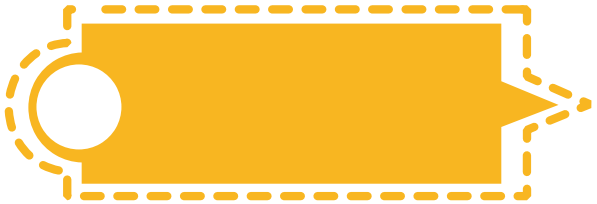
RELEASE NOTES

4



39.x

MediaWiki 1.39.7 (LTS)

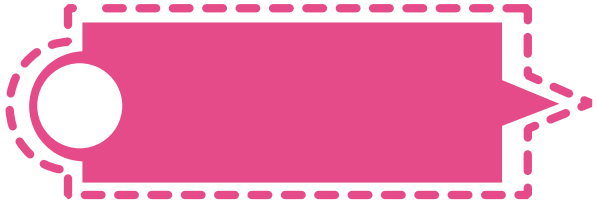


PHP 8.1.28

Fpm-fcgi module of Apache

MariaDB 10.3.39

Release date: 2023-05-10



RHEL 8 compatible

Enterprise Linux v8 Compatible
Rocky Linux release 8.9 (Green Obsidian)

New Features

- Only execute SMW Rebuild Data on demand
- Pretty URLs
- Deploy over local changes
- WebP image support
- Composer self-update
- Faster

Other Changes

- Reduce git clone size of MediaWiki by 2+GB / improve speed
- Add .editorconfig
- Version lock Ansible and Python
- Disable (automatic) Elasticsearch upgrades for predictability
- Set PHP-FPM default port to 9000 avoiding surprises or conflicts
- Syntax cleanup
- Bug fixes in Python, PHP, YAML
- Documentation of all 50+ meza.py functions
- Improve testing scripts
- Create CHANGELOG
- Remove obsolete $wgShellLocale
- Allow easier forking
- Enable Ansible debugger

Continuous Integration

We embraced the concept of using and updating the existing Meza integration tests. Specifically, there are three tests:

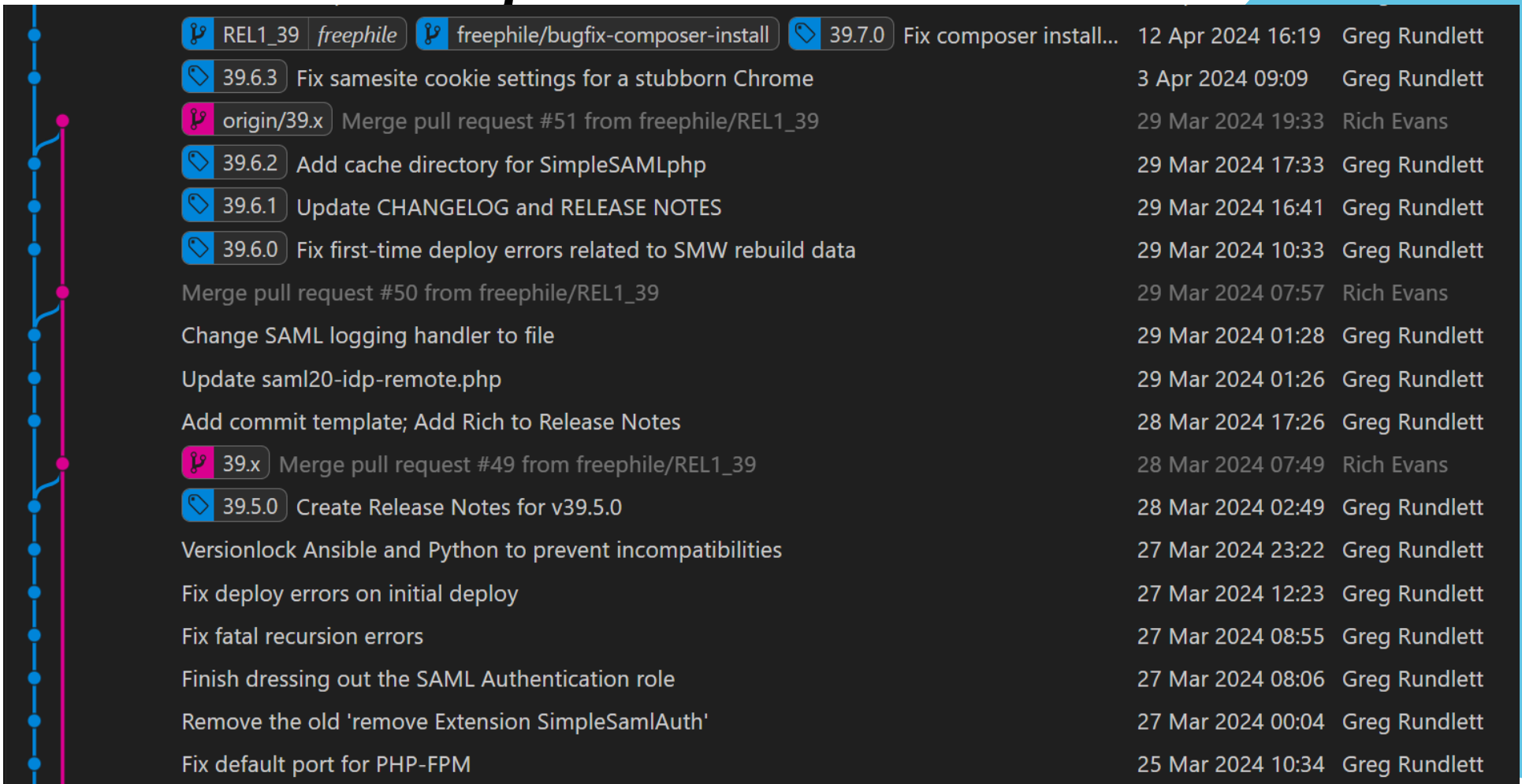
- image-check.sh
- server-check.sh
- wiki-check.sh

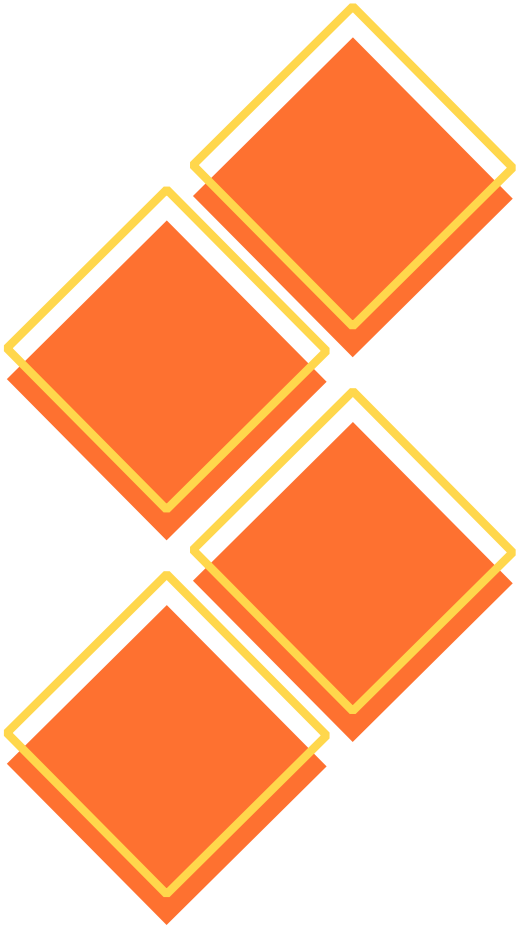
These integration tests are in turn used in system-level scripts such as

./tests/deploys/backup-to-remote.controller.sh

✅ This runs 70 tasks, including creating a new wiki.

Tags and Releases





Future

New Features
New Users
Updated Docs
Your needs

5

In the works

Live Stats package

Logstash
Kibana

Auth examples

Platform upgrades

Wider OS support

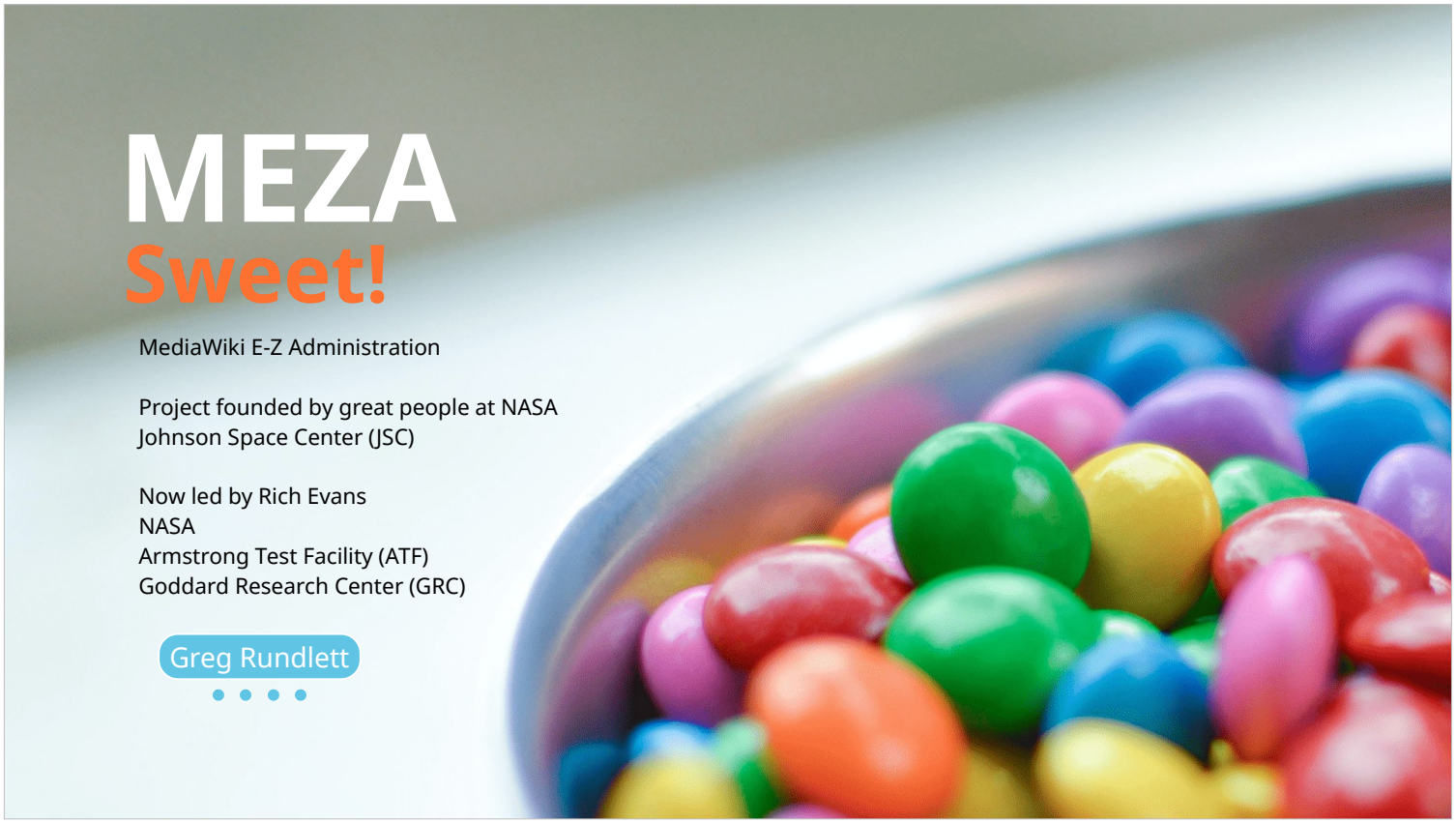
TLS automation

THANK YOU

CREDIT BY :

@lken__ @menggelinding_ @ealita.id @li_orch @cahyaning_asih

BIDADARI GIMPSCAPE



Welcome
My name is Greg Rundlett
I like to call myself a knowledge hacker
- someone at the intersection of free knowledge and free software.

Before I tell you about Meza, let me give you a little of the back-story and how I came to be here today.

I'm so thrilled to share the stage with Ward because my introduction to wikis was in 1999 using that first perl wiki as the backend at a startup company called Raging Bull

In 2013 after I lost my Mom to cancer and my job to outsourcing, I dared to start my own business called eQuality Technology offering free software solutions. By 2017, the business had pretty much focused on service and support of MediaWiki.

This happened because I discovered Meza and not only adopted it for my own use, but turned it into a product called QualityBox. I used the QualityBox system to service and support customer wikis. But, I couldn't scale the business as a solopreneur so gave up on the business in 2020 after the pandemic hit to go back to being an employee. I joined a multi-national software company with HQ in Cambridge, MA leading their adoption of MediaWiki – working "side-by-side" with their vendor Wikibase Solutions in the Netherlands.

With that background, let me tell you more about Meza.

In this short presentation, I’ll talk about WHAT Meza is

Who is Meza for?

Why should I care about Meza? (What’s in it for me?)

What is the current status of Meza

And we’ll close out with a look at what’s next for Meza.

Meza is

an Ansible-driven, Infrastructure as Code (IaC) complete platform for Wiki farms.

That’s the buzz-word compliant description.

Meza is a push-button solution to Knowledge Management.

That’s the sales & marketing description of what Meza is.

I like this visual description: The entire MediaWiki ecosystem in convenient spill-proof packaging.

That big planet in the middle is MediaWiki.

We've already heard about other MediaWiki "packages" or "MediaWiki+" like Canasta, BlueSpice, ProWiki or Fandom. I prefer the term "distribution" because these things are vendor-supplied, enhanced, and purpose-configured systems based on MediaWiki. I borrow the term distribution from the Drupal project

Meza is another distribution.

Language	Files	Lines	Blanks	Comments	Code	Complexity
YAML	125	9594	1272	1775	6547	0
Jinja	74	7832	1492	22	6318	280
Shell	26	1901	376	542	983	132
Alex	18	20	0	0	20	0
Plain Text	15	1366	244	0	1122	0
PHP	10	3747	636	486	2625	242
Markdown	7	942	249	0	693	0
Dockerfile	4	167	16	9	142	58
JavaScript	4	22950	1971	1307	19672	3170
SQL	3	913	114	4	795	0
JSON	2	218	0	0	218	0
Python	2	1841	26	100	1715	26
CSS	1	641	129	30	482	0
INI	1	1	0	0	1	0
License	1	22	5	0	17	0
gitignore	1	17	5	5	7	0
Total	**294**	**52172**	**6535**	**4280**	**41357**	**3908**

Estimated Cost to Develop (organic) **$2,783,081**
Estimated Schedule Effort (organic) 15.40 months
Estimated People Required (organic) 7.77

Technically speaking, Meza is a bunch of YAML, Jinja templates, shell scripts, Python, PHP, Markdown, and Docker files.

According to a reliable formula, using the average developer salary for 2023, **Meza is a complex and powerful software system worth approximately $2.8 million dollars.**

average salary = $116,393/year, overhead = 2.40 The table was generated using the Sloc, Cloc and Code (scc) tool https://github.com/boyter/scc

There are MediaWiki consultants here,
System Administrators responsible for their organizations' wikis,
Hosters who offer MediaWiki as a Service

Meza is for you.

Meza is not for hobbyist. Meza is not for beginners. Meza is complex. MediaWiki is complex.

Running your own MediaWiki because it's free software is like setting up your own Mail Server to do email. No amount of automation is going to eliminate the underlying fact that this is a tool to basically orchestrate the entire web.

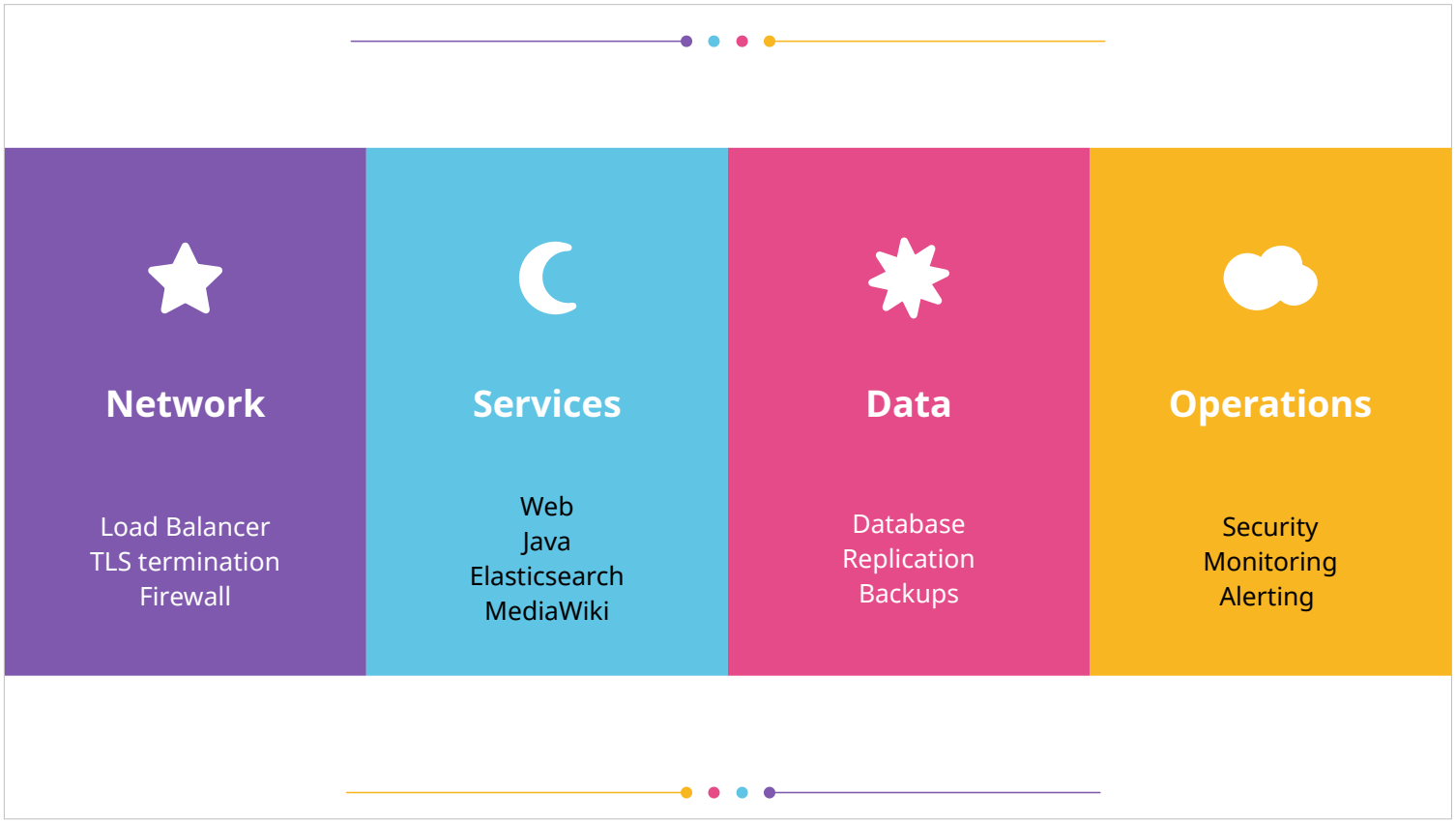
If you want a wiki, but don't do this stuff for a living, please seek professional help – where plans range from white glove service, to extremely affordable and even "free".

But why?

“Because it’s good for ‘ya”

What do you mean it’s good for me?

If it were a food product instead of a technical one, I’d put it like this:



From a technical perspective, Meza takes care of the full spectrum of needs:

Network
Services
Data
and Operations

Tasks / Responsibilities

- Operating System
- Apache + modules
- PHP + modules
- Composer +
- Python
- Automation
- Backups
- MediaWiki

- Code updates
- Database updates
- Configure wikis
- Create new wikis
- Delete wikis
- Task runner
- Elasticsearch
- Firewall

- Distributed filesystem
- Proxy
- Web Content
- Image processing
- Security
- Lua
- Cache

- Logging
- Monitoring
- Cluster sync
- SAML / Auth

Really? Meza does all that? What responsibilities does it handle?

Extensions

AdminLinks	CreateUserPage	Graph	ModernTimeline
AdvancedSearch	CSS	HeaderFooter	MultimediaViewer
Arrays	DataTransfer	HeaderTabs	MyVariables
Bootstrap	DeleteBatch	ImageMap	Network
CharInsert	DismissableSiteNotice	InputBox	NumerAlpha
CirrusSearch	DisplayTitle	Interwiki	OpenLayers
Cite	DynamicSidebar	LabeledSectionTransclusion	
CodeEditor	Echo	Lockdown	(continued)
CodeMirror	Elastica	Maps	
CollapsibleVector	ExternalData	MediaFunctions	
CommentStreams	FlexDiagrams	Mermaid	
ContributionScores	Flow		

And, as we saw in Yaron's talk about selecting quality extensions. Meza is opinionated about – and saves you the effort of – selecting the baseline extensions that you need to make a MediaWiki Plus

Extensions

PageForms	ExtraSpecialProperties	SubPageList	VisualEditor
PageImages	Semantic MediaWiki	SyntaxHighlight_GeSHi	WatchAnalytics
ParserFunctions	Semantic Result Formats	TemplateData	WhitelistPages
PipeEscape	Semantic Scribunto	TextExtracts	WhoIsWatching
ReplaceText	SimpleBatchUpload	Thanks	WhosOnline
RevisionSlider	SimpleMathJax	LanguageSelector	Widgets
Scribunto	SubpageFun	Variables	WikiEditor
CompoundQueries		VEForAll	YouTube
SemanticDrilldown			

Plus

So Meza preinstalls curated MediaWiki extensions.

In a minute, I'll talk about how Meza give you extreme flexibility to add your own selections.

Here’s what zero to 100 looks like with Meza

Here’s a more linear view.

1)Download from GitHub
2)Deploy from your laptop on the beach
3)Create a ‘conf’ repo to store and track your customizations
4)Continuously deploy to any target:
    1)Local development (VirtualBox)
    2)Any tier in your Cloud / Onsite Infrastructure
        1)DEV
        2)QA / Staging
        3)PROD

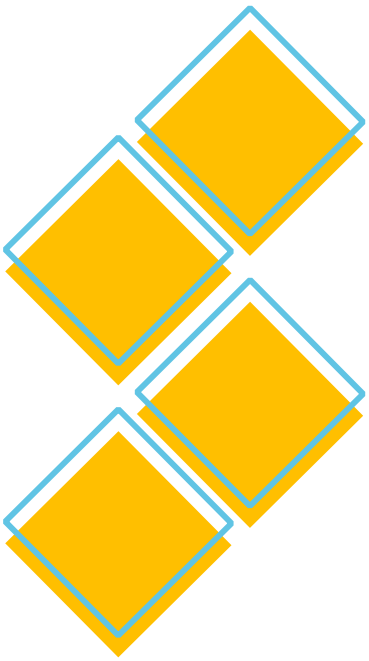
Like Ani DiFranco sings:

> ”I am 32 flavors and then some”

https://youtu.be/vVg7mtgEqGY?t=172

One of Meza’s strongest features is that it allows for complete overrides and selections for each wiki in the farm. So, like going to Baskin Robbins for ice cream, everyone gets their own choices instead of the one carton of ice cream that Dad likes.

If I use another food metaphor, Meza has all the essential amino acids.



Status

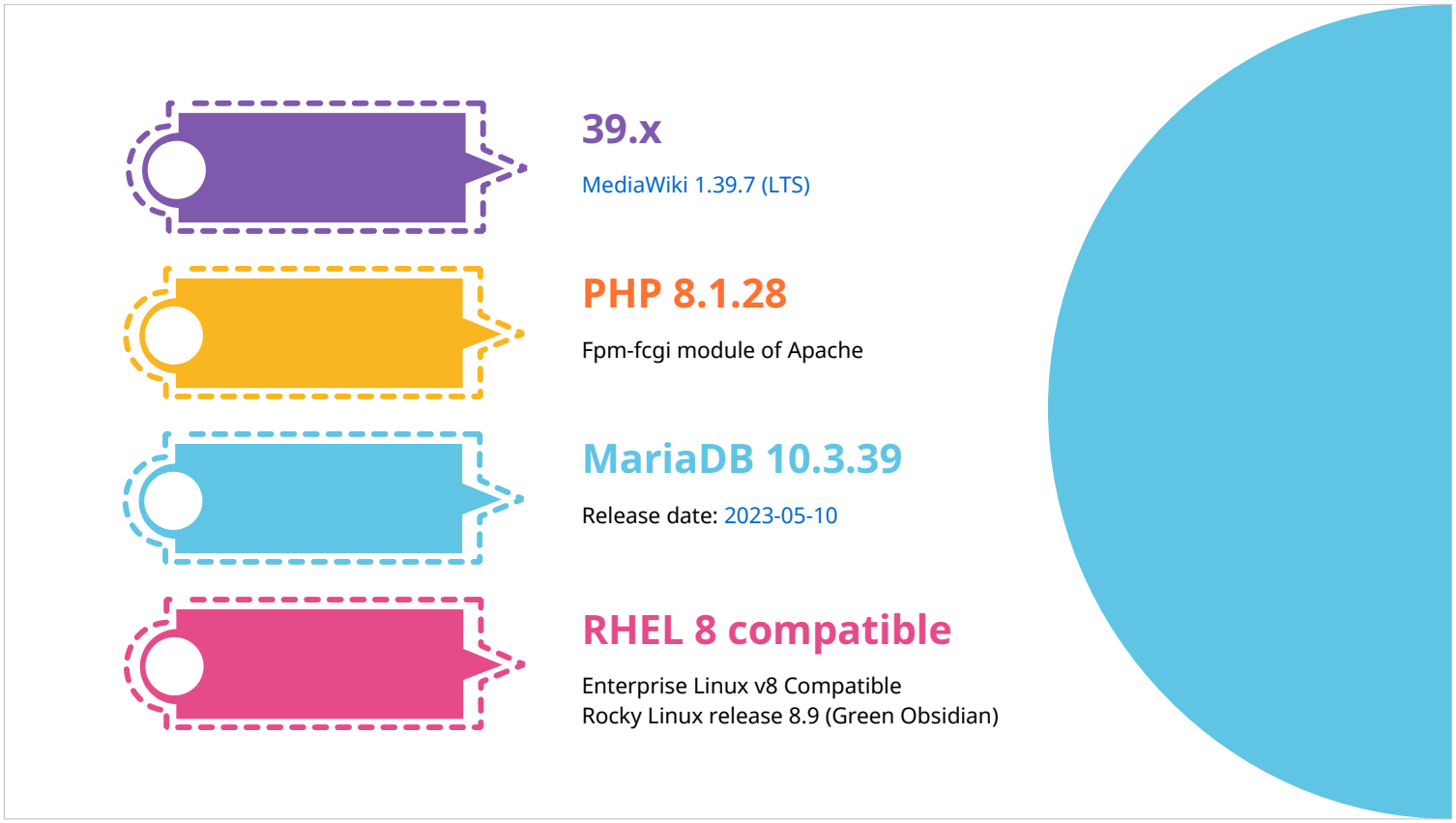
Hot, out of the oven

RELEASE NOTES

4

After a period of relative stagnation for the Meza originally developed at JSC in Houston, TX, the ATF/GRC facilities in Ohio hired me through affiliation with Bryan Hilderbrand and Yaron Koren to revive the project.

- MediaWiki REL1.39 upgrade from REL1_35
- PHP 8.1 compatibility
- And pretty URLs



MediaWiki 1.39.6
Semantic MediaWiki 4.1.3
PHP 8.1 using the Remi repo
Elasticsearch 7.10.2
SAML Authentication

- SimpleSAMLphp library to 2.2.1
- Extension Pluggable Auth 7.0
- Extension SimpleSAMLphp 7.0

Composer

New Features

- Only execute SMW Rebuild Data on demand
- Pretty URLs
- Deploy over local changes
- WebP image support
- Composer self-update
- Faster

meza deploy monolith --tags smw-data
meza deploy monolith --tags smw-data,search-index

There is no index.php in URLs, just mysite.com/wiki/SomePage Easier to type, easier to read, easier to remember, shorter, and pretty!

overwrite_local_git_changes

Other Changes

- Reduce git clone size of MediaWiki by 2+GB / improve speed
- Add .editorconfig
- Version lock Ansible and Python
- Disable (automatic) Elasticsearch upgrades for predictability
- Set PHP-FPM default port to 9000 avoiding surprises or conflicts
- Syntax cleanup
- Bug fixes in Python, PHP, YAML
- Documentation of all 50+ meza.py functions
- Improve testing scripts
- Create CHANGELOG
- Remove obsolete $wgShellLocale
- Allow easier forking
- Enable Ansible debugger

Meza is not a WMF project, but we used WMF-style quality control and continuous integration techniques in the process (the kind you get when you do use Gerrit for your source code).

And used industry best practices in the Ansible Automation world.

We performed static analysis with PHPStan and Phan on 92 extensions (55 without phan config and 37 with).

Introduced
Editor config
Yamllint
Ansible Lint

Continuous Integration

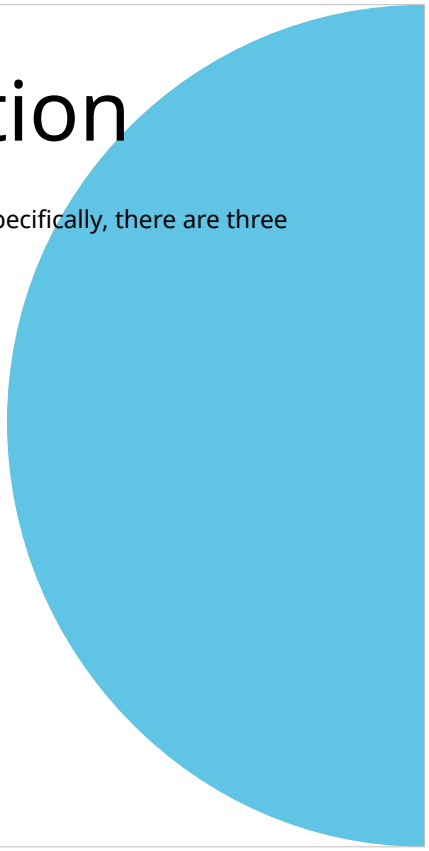
We embraced the concept of using and updating the existing Meza integration tests. Specifically, there are three tests:

- image-check.sh
- server-check.sh
- wiki-check.sh

These integration tests are in turn used in system-level scripts such as

./tests/deploys/backup-to-remote.controller.sh

✅ This runs 70 tasks, including creating a new wiki.



we embraced the concept of using and updating the existing Meza integration tests. Specifically, there are three tests:

- image-check.sh
- server-check.sh
- wiki-check.sh

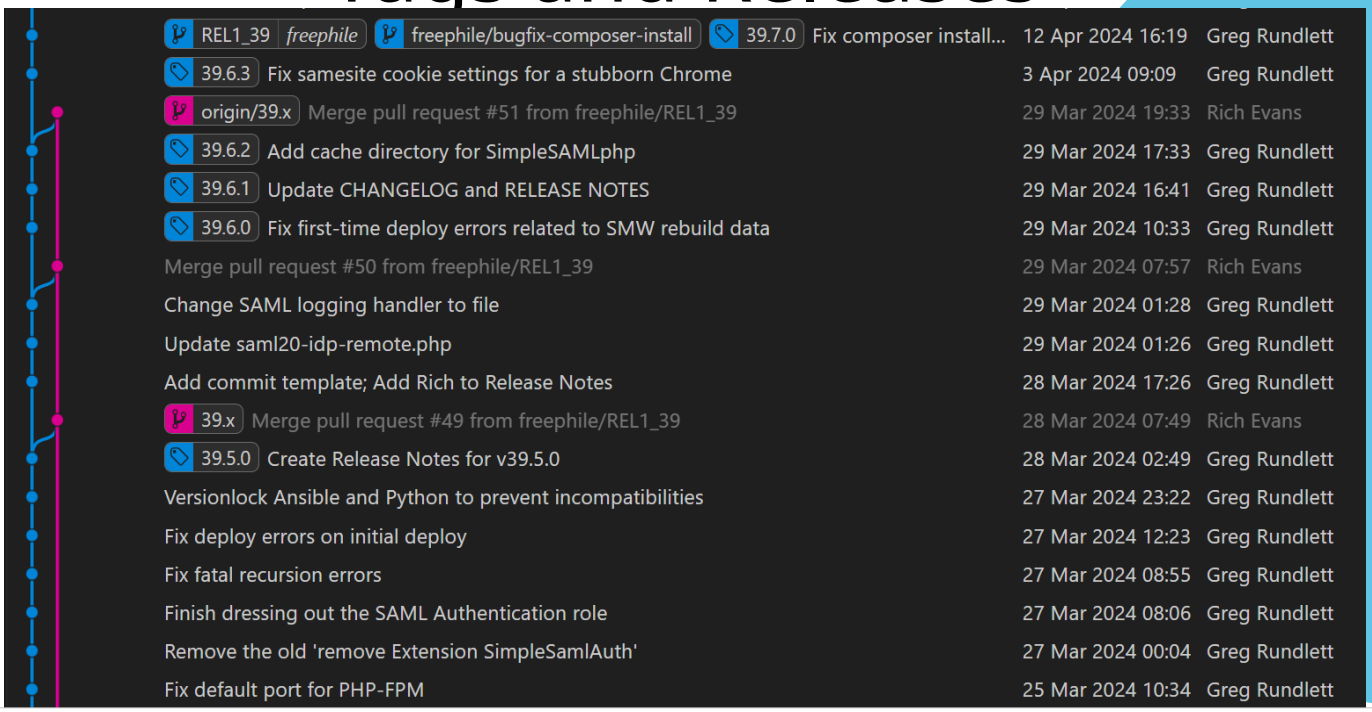
These integration tests are in turn used in system-level scripts such as ./tests/deploys/backup-to-remote.controller.sh

This script does a fairly good job of testing the whole system. In fact, all of these scripts were meant to emulate the Continuous Integration operations of Travis CI which was initially used by the project (when Travis CI was freely available to open source projects.)

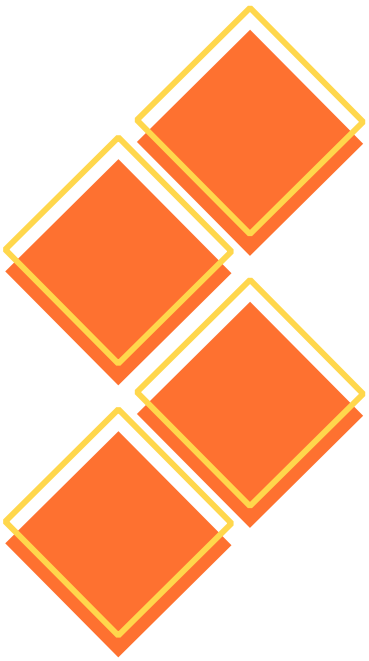
After some development, we were able to run a successful
✅ bash ./tests/deploys/backup-to-remote.controller.sh monolith

(This runs 70 tasks, including creating a new wiki.)

Tags and Releases



This slide just shows that we use Semantic Versioning and Git tags in the development and release process.



Future

New Features
New Users
Updated Docs
Your needs

5

So, Meza is ready for immediate use if you want to enjoy a “push-button experience for Knowledge Management”

But, there’s a lot left to do.

If I’m lucky, someone (nudge nudge) will pay me to carry the mission forward.

My ideas about what Meza needs are in my fork https://github.com/freephile/meza/issues (59 issues)

The 31 closed issues dig deeper into what happened in the upgrade to REL1_39

One thing I’d like to see is for continued consolidation and collaboration between “distributions” whether pure open source or from commercial vendors.

I especially want to thank Rich Evans at NASA for giving me the opportunity to carry the ball from REL1_35 to REL1_39

https://www.linkedin.com/in/gregrundlett