



How to be part of the MediaWiki developer community

Ryan Lane <ryan@wikimedia.org>

Wikimedia Foundation Inc.

Focus of this talk

- Why to participate
- Where to find resources
- Methods of communication
- How to engage the community
- Our philosophy
- How to code for Wikimedia sites



Why to participate

- Mentor and be mentored
- Build a strong reputation
- Have a public work record
- Support an awesome mission



Where to find resources

- mediawiki.org
- wikitech.wikimedia.org
- Mailing list archives
- These slides
- Me, after the talk



How to engage the community

- Discuss
- Commit
- Participate



How to engage the community

- Document
- Communicate changes
- Don't be afraid of a language barrier
 - Be an ambassador, if possible



Our philosophy

- Engage early
- Release early, release often
- Scratch your own itch



Coding for WMF: Security

- Security is important. *Really.*
- People rely on developers to write secure code, so:
 - An insecure extension in SVN...
 - An insecure extension on Wikipedia...



Common vulnerabilities to avoid

- SQL injection
- Cross site scripting (XSS)
- Cross site request forgery (CSRF)
- Register Globals



SQL injection

Problem:

```
$sql = "INSERT INTO Students VALUES ( $name, ... );";  
INSERT INTO Students VALUES ( 'Robert' ); DROP TABLE  
Students; --', ... );
```

Fix:

```
INSERT INTO Students VALUES ( 'Robert\''); DROP TABLE Students;  
--' , ... );
```



Using MediaWiki's functions

Unacceptable:

```
$dbr->query( "SELECT * FROM foo WHERE foo_id=' $id'" );
```

Acceptable:

```
$escID = $dbr->addQuotes( $id ); $dbr->query( "SELECT * FROM foo  
WHERE foo_id= $escID" );
```

Correct:

```
$dbr->select( 'foo', '*', array( 'foo_id' => $id ) );
```



Cross Site Scripting (XSS)

Problem:

```
$val = $wgRequest->getVal( 'input' ); $wgOut->addHTML( "<input  
type=\"text\" value=\" $val\" />" );
```

Fix:

```
value="&lt;script&gt;evilStuff();&lt;/script&gt;"
```



Using MediaWiki's functions

Unacceptable:

```
$html = "<input type=\"text\" name=\"foo\" value=\" $val\" />";
```

Acceptable:

```
$val = htmlspecialchars( $val ); $html = "<input type=\"text\"  
name=\"foo\" value=\" $val\" />";
```

Correct:

```
$html = Html::input( 'foo', $val );
```



Cross Site Request Forgery (CSRF)

Problem:

```
<form id="myForm" method="POST" action="....."> <input
type="hidden" name="title" value="Foo" /> <input type="hidden"
name="action" value="delete" /> <input type="hidden"
name="wpReason" value="MUAHAHA" />
</form> <script>
$( '#myForm' ).submit(); </script>
```

Fix:

```
$html .= Html::hidden( 'token', $wgUser->editToken() ); ... if (
!$wgUser->matchEditToken( $token ) ) {
// refuse edit
```



General notes on security

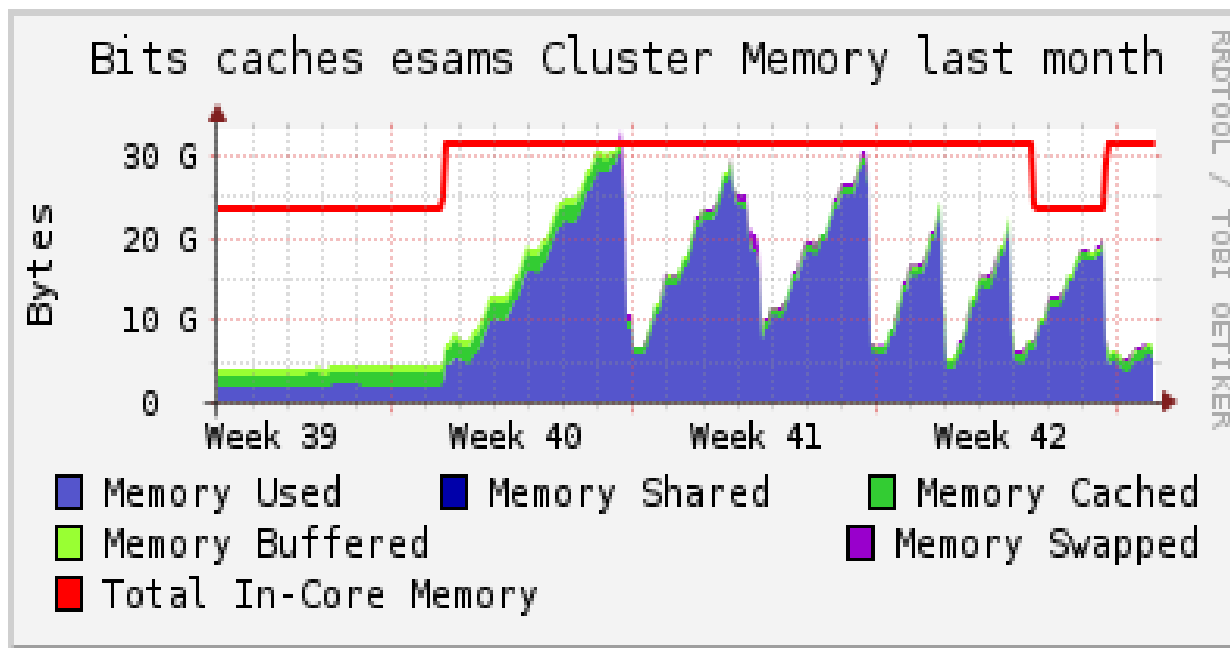
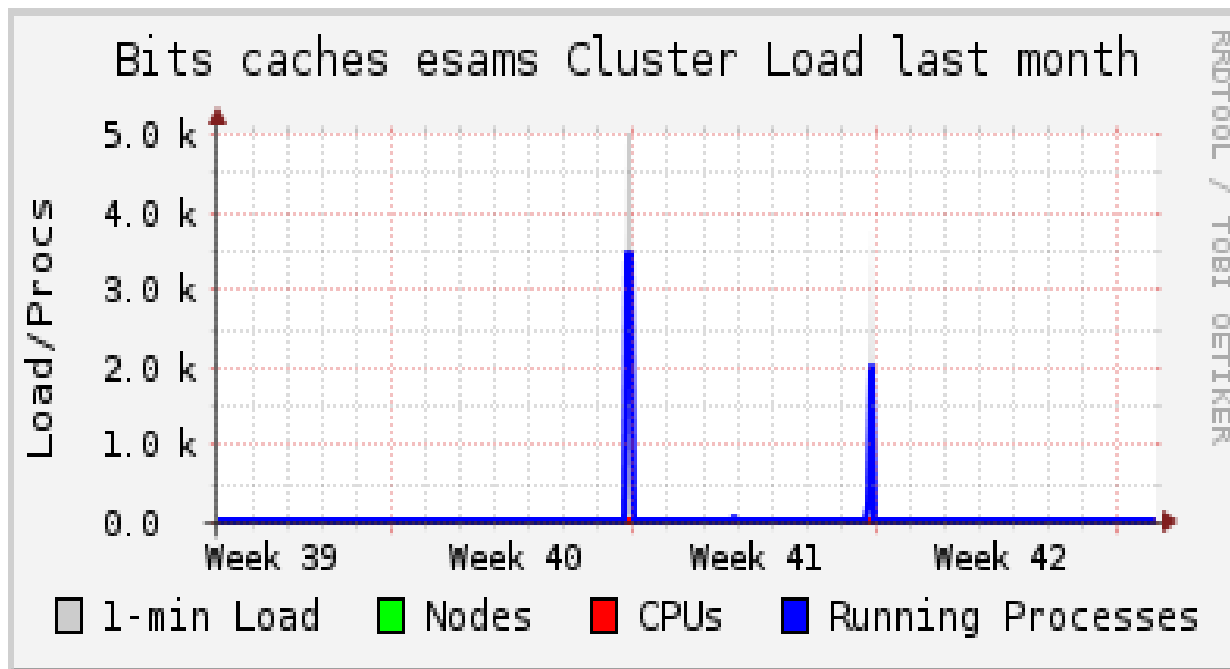
- Don't trust *anyone*
- Sanitize all input
- Write code that is *demonstrably secure*
- Best of all: try to break and hack your own code



Coding for WMF: Scalability and performance

- Wikimedia sites are huge
 - 5th most visited web presence
- Code must be:
 - Performant
 - Scalable





Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.
 How to be part of the MediaWiki developer community



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.
How to be part of the MediaWiki developer community

Coding for WMF: Scalability and performance

- Cache
- Optimize
- Profile



MediaWiki Profiling

<http://noc.wikimedia.org/cgi-bin/report.py>

[\[zhwiki\]](#) [\[thumb\]](#) [\[dewiki\]](#) [\[bigpage\]](#) [\[enwiki\]](#) [\[others\]](#) [\[flaggedrevs\]](#) [showing 50 events, [show more](#)]

name	count	cpu%	cpu/c	real%	real/c
PPFrame_DOM::expand	2777300471	409	1.8	322	1.89
Parser::braceSubstitution	478045780	340	8.66	266	9.07
-total	7216450	100	169	100	226
Parser::braceSubstitution-pfunc	453314242	97.8	2.63	76.5	2.75
MediaWiki::performRequestForTitle	3445759	74.4	263	71.7	339
Parser::internalParse	6879781	78.4	139	61.7	146
Parser::replaceVariables	76312501	71.4	11.4	56.4	12
MediaWiki::performAction	1329950	65.8	604	54.8	671
Parser::parse	3242613	65.5	246	52.4	263
Article::view	956646	57.6	735	46.8	797
Parser::braceSubstitution-setup	478043151	53.9	1.38	41.9	1.43
Parser::parse-Article::getOutputFromWikitext	304682	49.8	1.99e+03	39.6	2.11e+03
Parser::argSubstitution	690835928	46	0.813	36.4	0.858
api.php	3720263	13.1	42.8	17.9	78.4
API:main	3720255	12.8	41.8	17.6	77.2



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.
How to be part of the MediaWiki developer community

Coding for WMF: Concurrency

- Assume a clustered architecture, *always*
- Your code will run concurrently
- It can result in strange bugs



Closing notes

- This talk is an incomplete guideline
- Being a MediaWiki developer is hard
 - But it is very rewarding!
- Communication is **very** important
 - Ask the experts!



Questions / comments?

- E-mail: Ryan Lane <ryan@wikimedia.org>
- IRC: Freenode: #mediawiki, #wikimedia-dev, #wikimedia-tech
- Mailing lists: mediawiki-l, wikitech-l



Communication resources

- Mailing lists
 - http://www.mediawiki.org/wiki/Mailing_lists
 - Important lists:
 - mediawiki-l: A MediaWiki support list
 - wikitech-l: A MediaWiki developer's list
 - mediawiki-api: A MediaWiki developer's list for the API
- IRC channels (on freenode)
 - #mediawiki: A MediaWiki support channel
 - #wikimedia-dev: A MediaWiki developer's channel



Developer resources

- http://www.mediawiki.org/wiki/Developer_hub/ja - developer hub
 - Developer hub: lists resources, guidelines, and code documentation
- http://www.mediawiki.org/wiki/How_to_become_a_MediaWiki_hacker/ja
 - How to become a MediaWiki hacker: introduction into how to do MediaWiki development
- http://www.mediawiki.org/wiki/Security_for_developers
 - Security for developers: essential security documentation



Developer resources

- http://www.mediawiki.org/wiki/Manual:Coding_conventions/ja
 - Coding conventions: conventions required for all Wikimedia run software
- <http://www.mediawiki.org/wiki/Localisation/ja>
 - Localisation: resources to write code that can be easily localised
- http://www.mediawiki.org/wiki/Code_review_guide
 - Code review guide: how your code will be reviewed before inclusion

