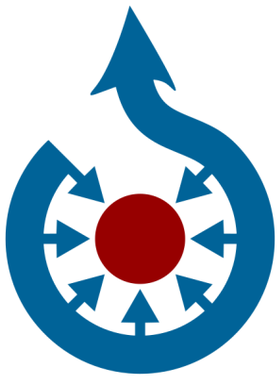
Wikimedia architecture

Ryan Lane <ryan@wikimedia.org>
Wikimedia Foundation Inc.



Overview

- Intro
- Our technical operations
- Global architecture
- Application servers
- Caching
- Storage
- Load balancing
- Content Delivery Network (CDN)

Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Top five worldwide sites

Company	Users	Revenue	Employees	Server count
Google	920 million	$23 billion	20,600	1,000,000+
Microsoft	740 million	$58 billion	93,000	50,000+
YAHOO!	600 million	$6 billion	13,900	50,000+
facebook	470 million	$300 million	1,200	30,000+
WIKIMEDIA	350 million	$20 million	50	350

Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.
Wikimedia architecture

Our operations

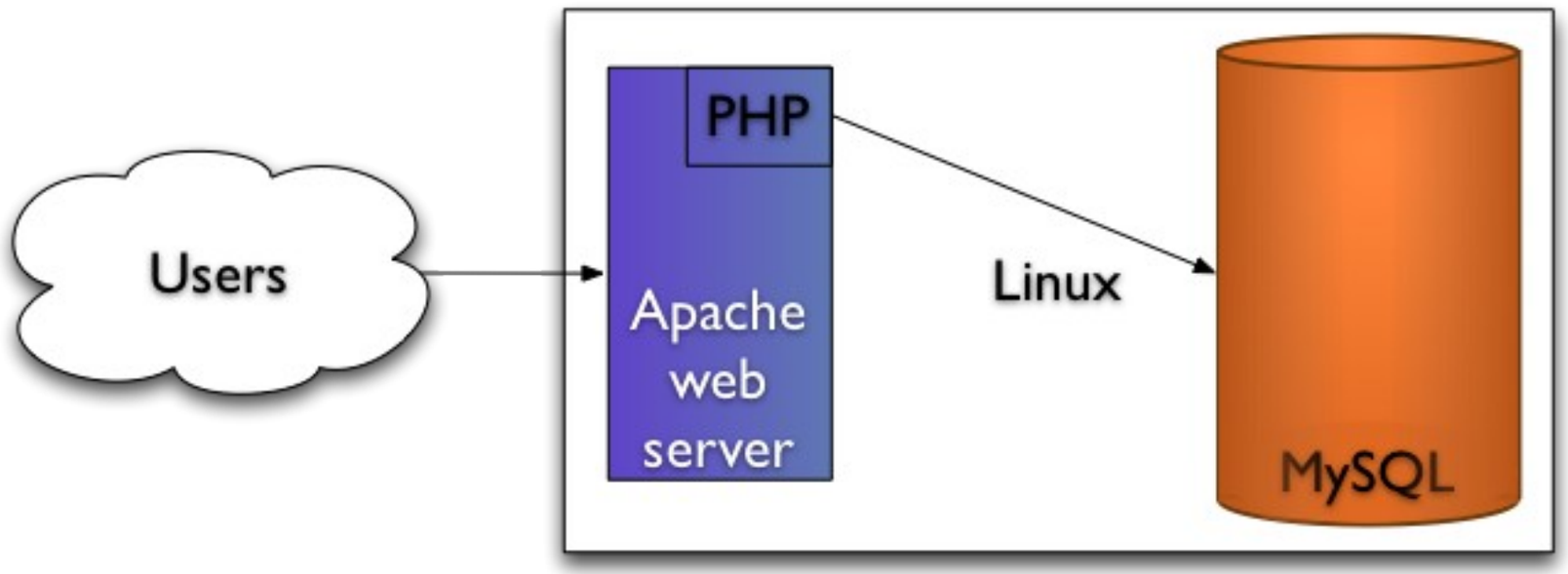
- Currently managed by ~6 ops engineers
- Historically ad-hoc, “fire fighting mode”
- Technical staff spread out globally
- Always someone awake...but no on-call
- Working to engage community operations contributors

Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

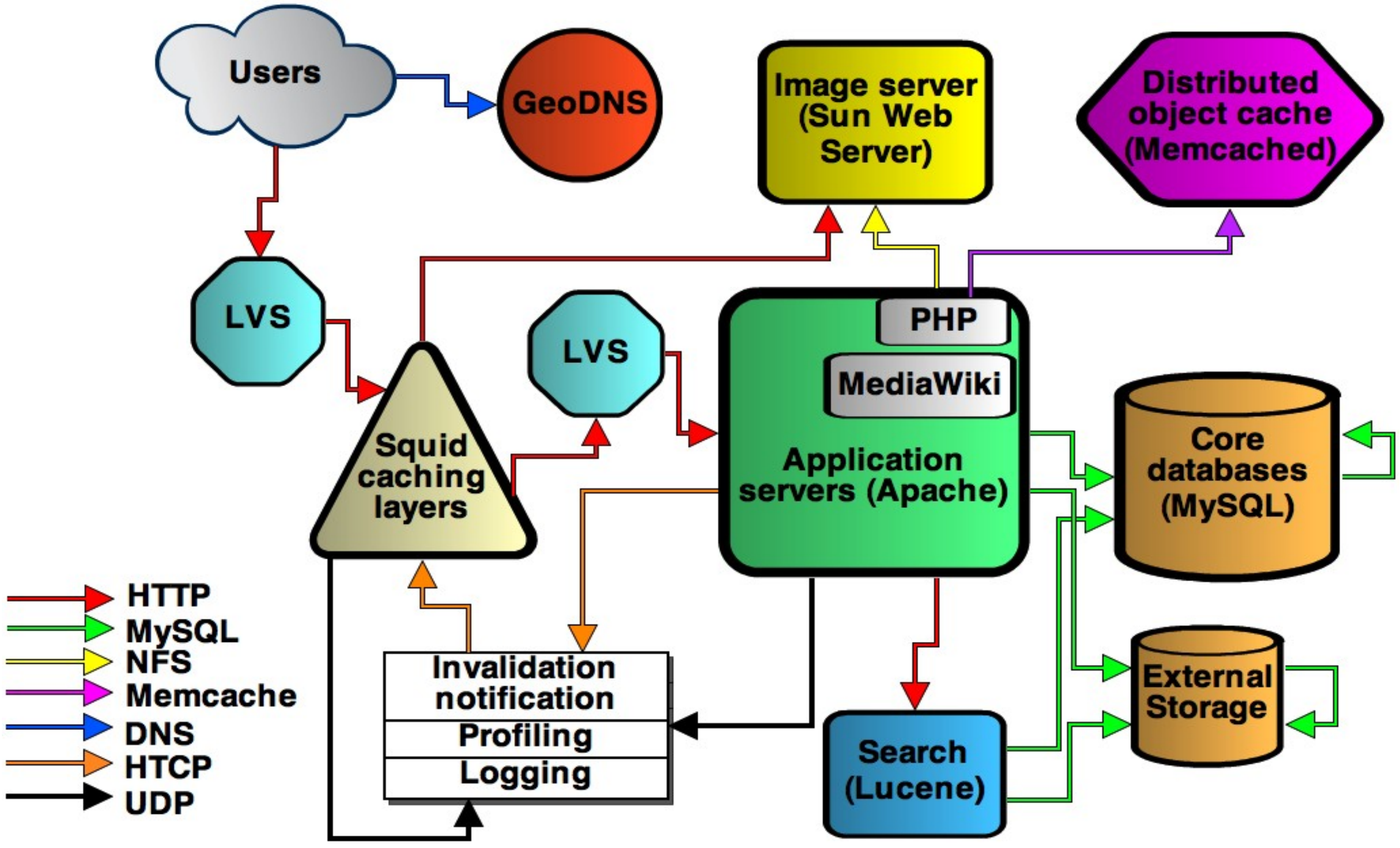
Operations communication

- Most communication public on IRC
- Documentation in a wiki (http://wikitech.wikimedia.org)
- Sensitive communication via private lists and resource trackers

Architecture: LAMP...

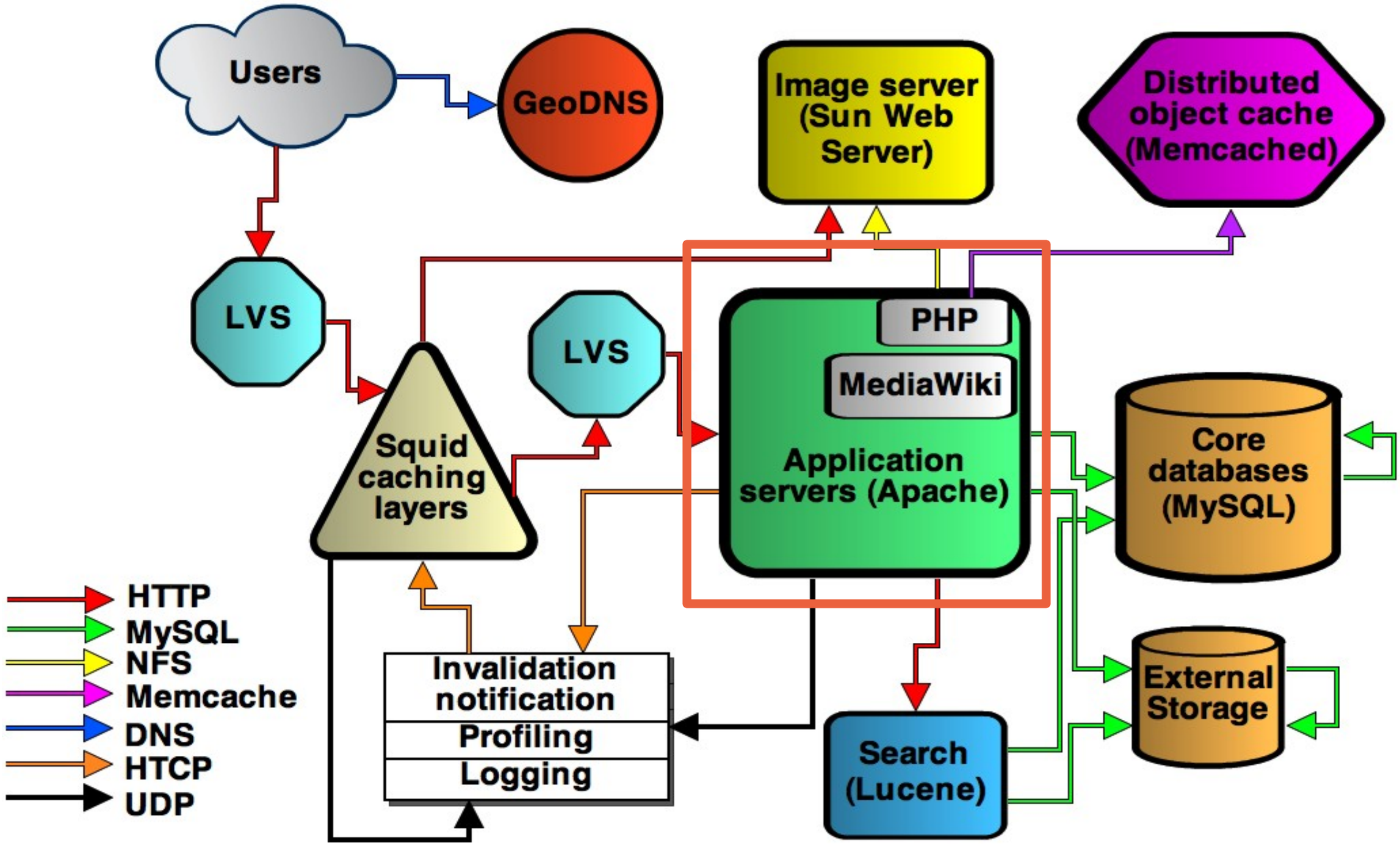


...on steroids.



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

The Wiki software

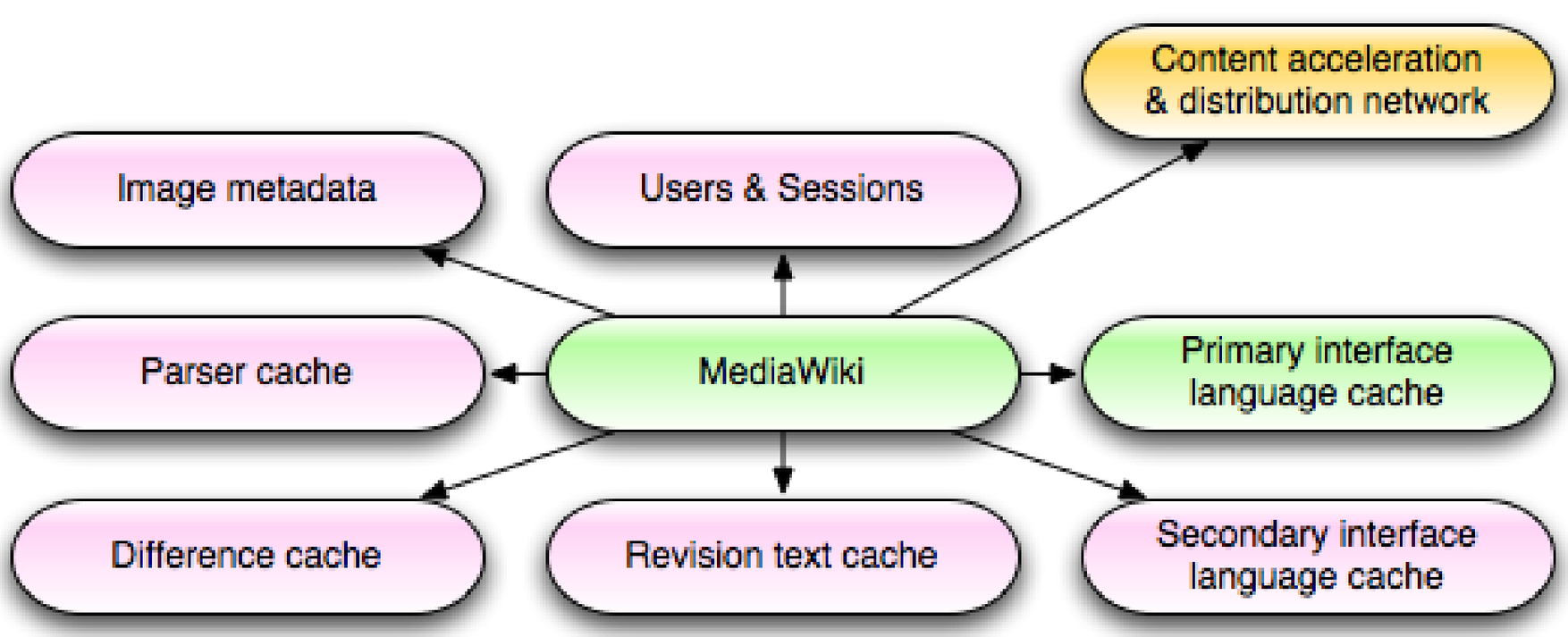
- All Wikimedia projects run on a MediaWiki platform
- Designed primarily for Wikimedia sites
- Very scalable, very good localization
- Open Source PHP software (GPL)

MediaWiki optimization

- We try to optimize by...
  - not doing anything stupid
  - caching expensive operations
  - focusing on the hot spots in the code (profiling!)
- If a MediaWiki feature is too expensive, it doesn't get enabled on Wikipedia

MediaWiki caching

- Caches everywhere
- Most of this data is cached in Memcached, a distributed object cache



MediaWiki profiling

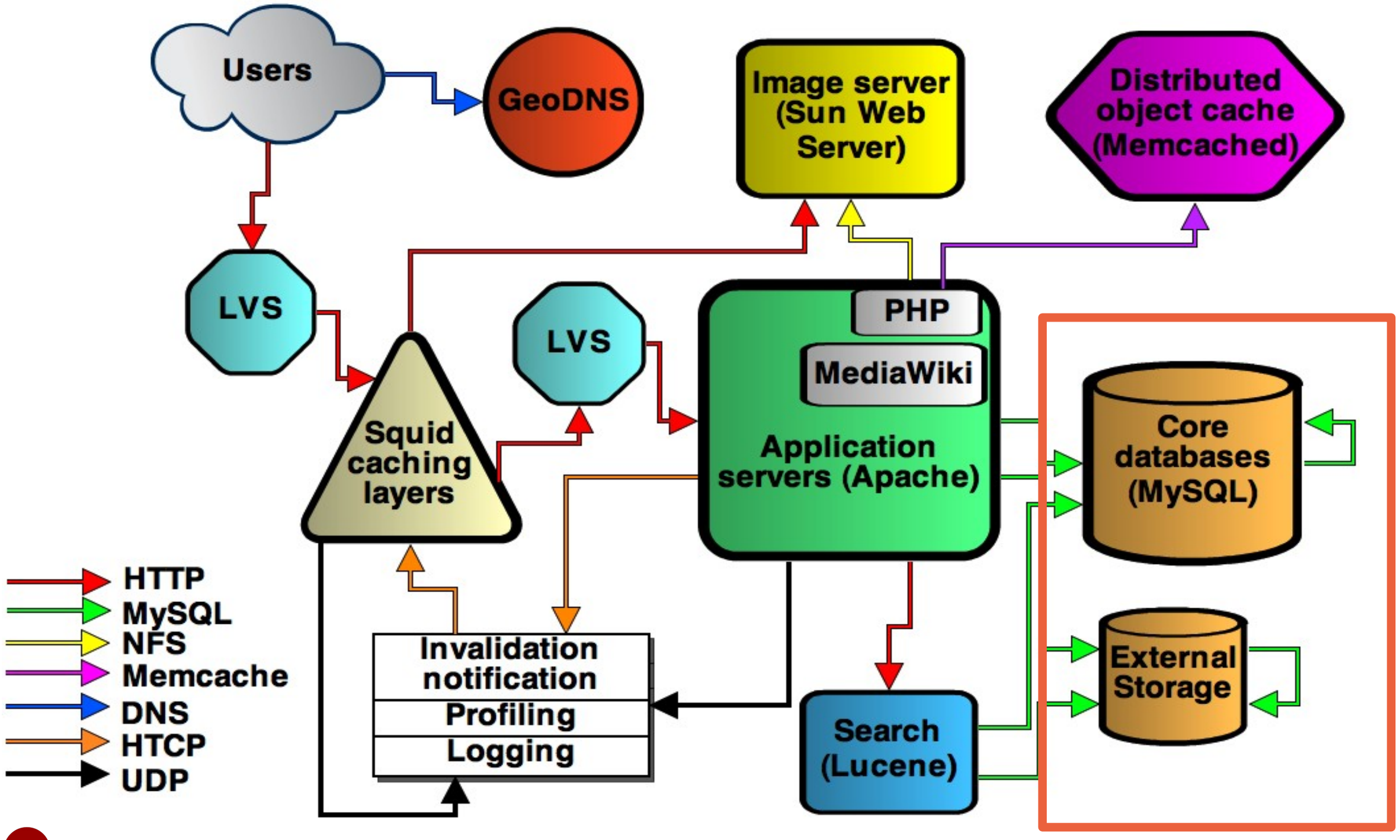
http://noc.wikimedia.org/cgi-bin/report.py

[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.

Wikimedia architecture

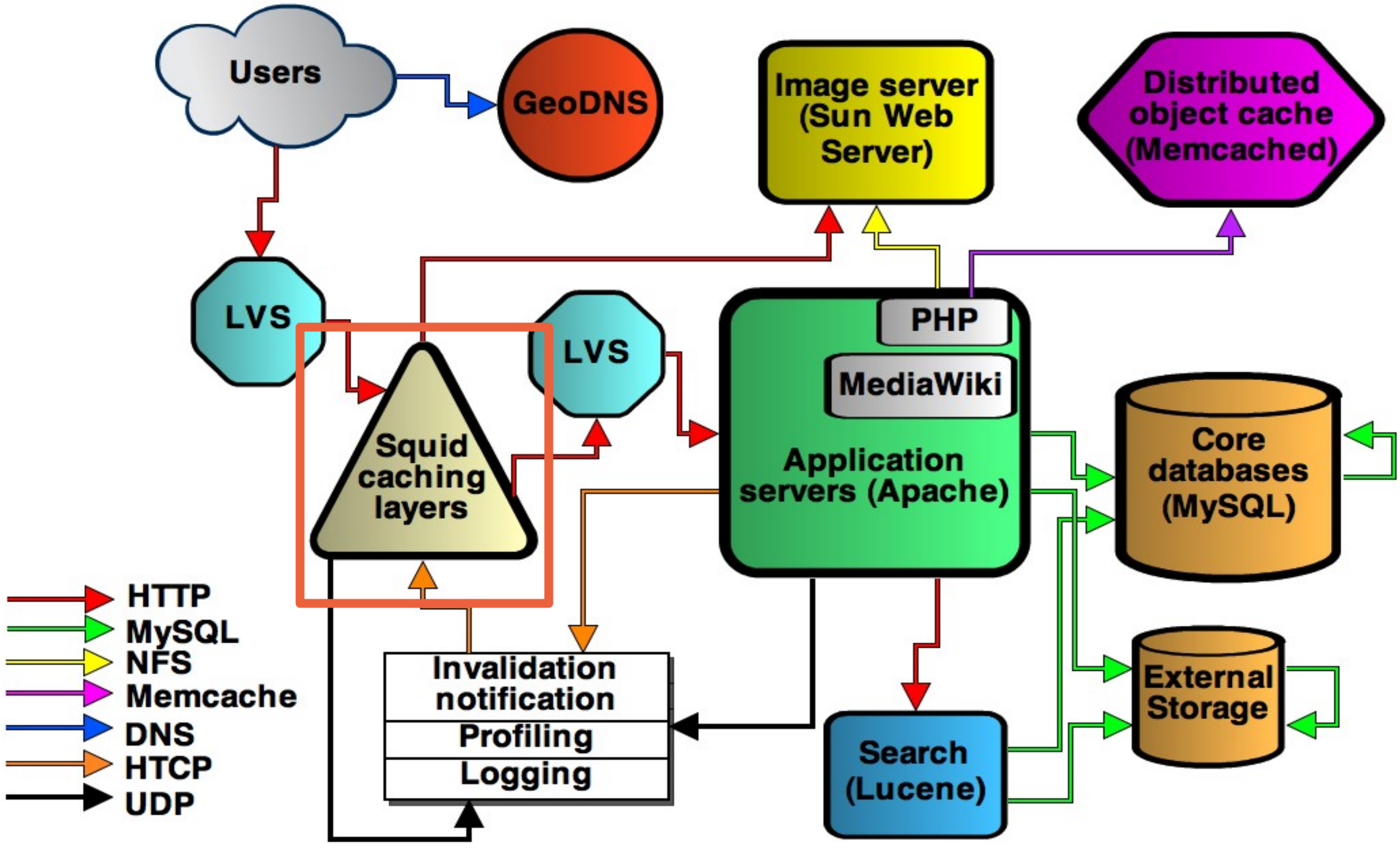


Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

Core databases

- One master, many replicated slaves
- Load balanced reads to slaves, write operations to master
- Separate database per wiki
- Separate big, popular wikis from smaller wikis (sharding)



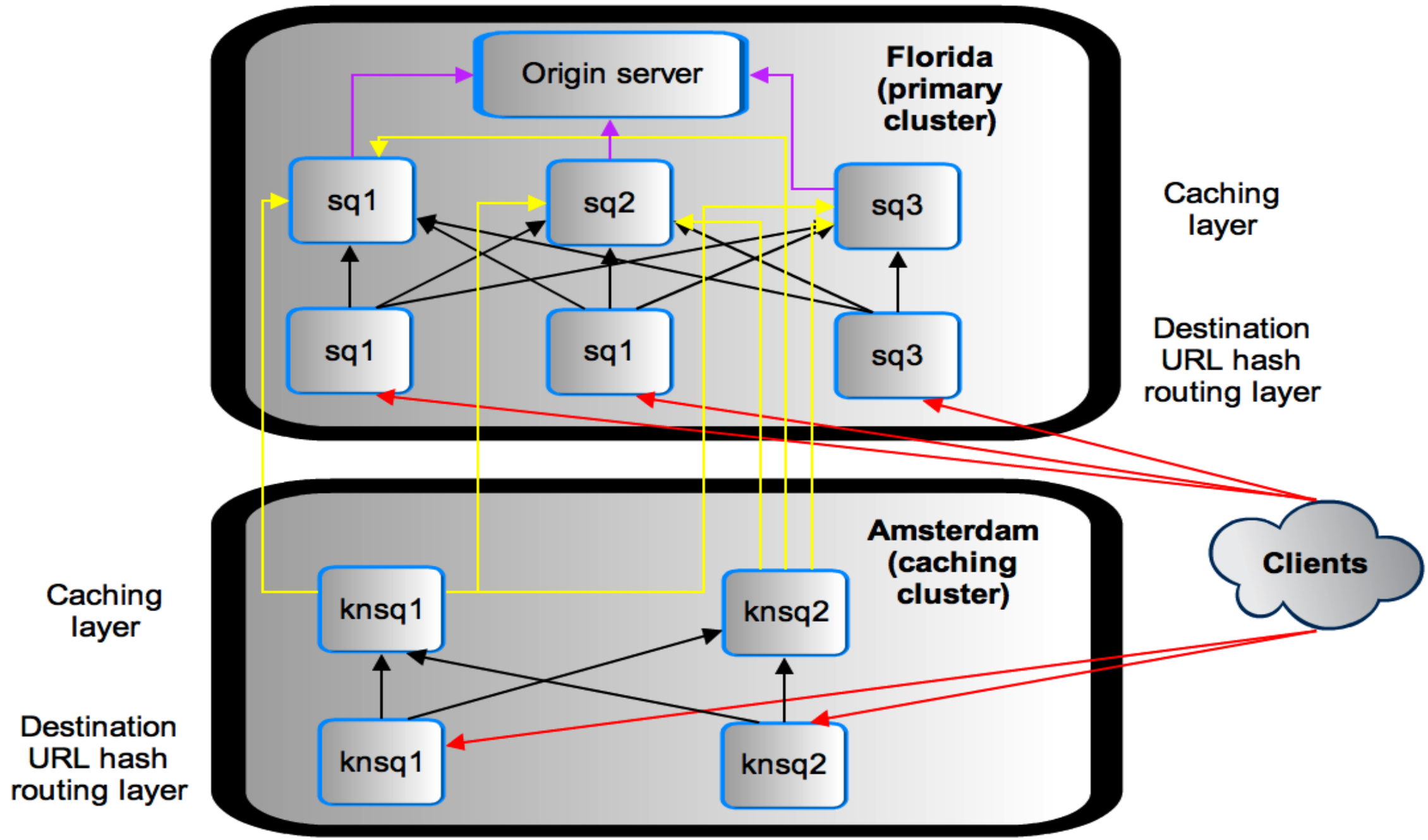
Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

Squid caching

- Caching reverse HTTP proxy
- Serves most of our traffic
- Split into three groups
- Hit rates: 85% for Text, 98% for Media, since the use of CARP

CARP



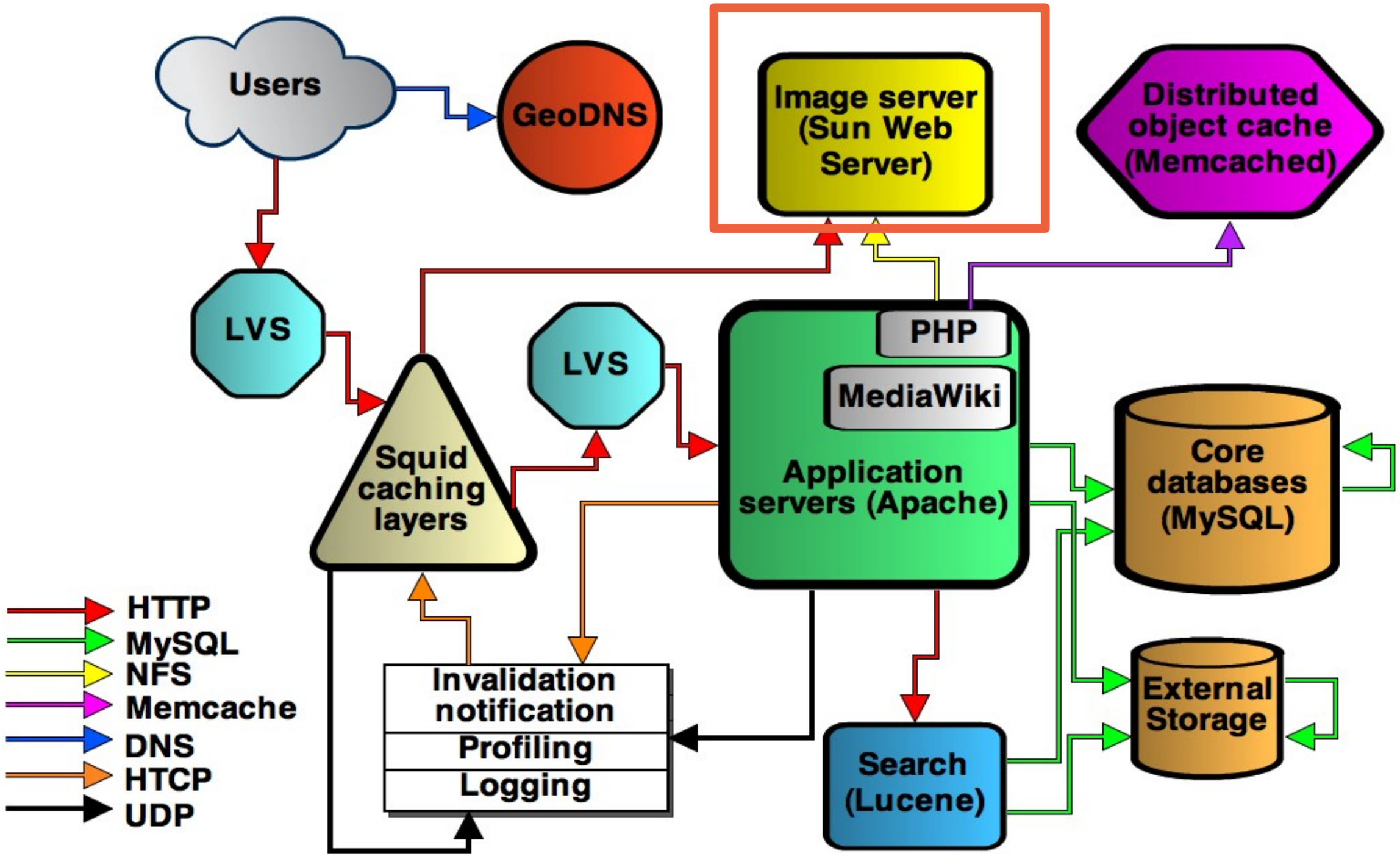
Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Squid cache invalidation

- Wiki pages are edited at an unpredictable rate
- Users should always see current revision
- Invalidation through expiry times not acceptable
- Purge implemented using multicast UDP based HTCP protocol

Varnish Caching

- Currently used for delivering static content such as javascript and css files (bits)
- Will eventually replace the Squid architecture
- 2-3 times more efficient than Squid



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

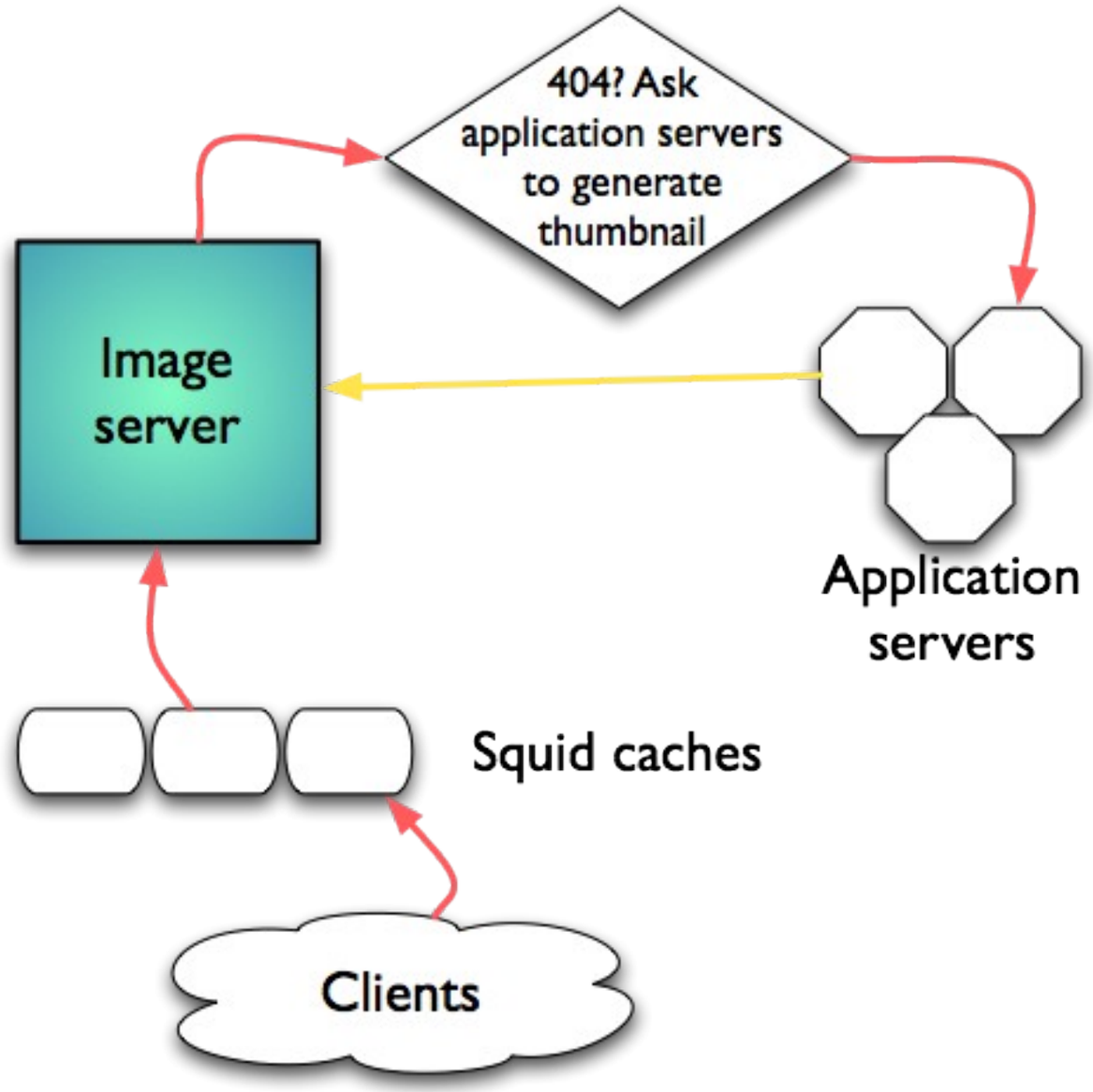
Media storage

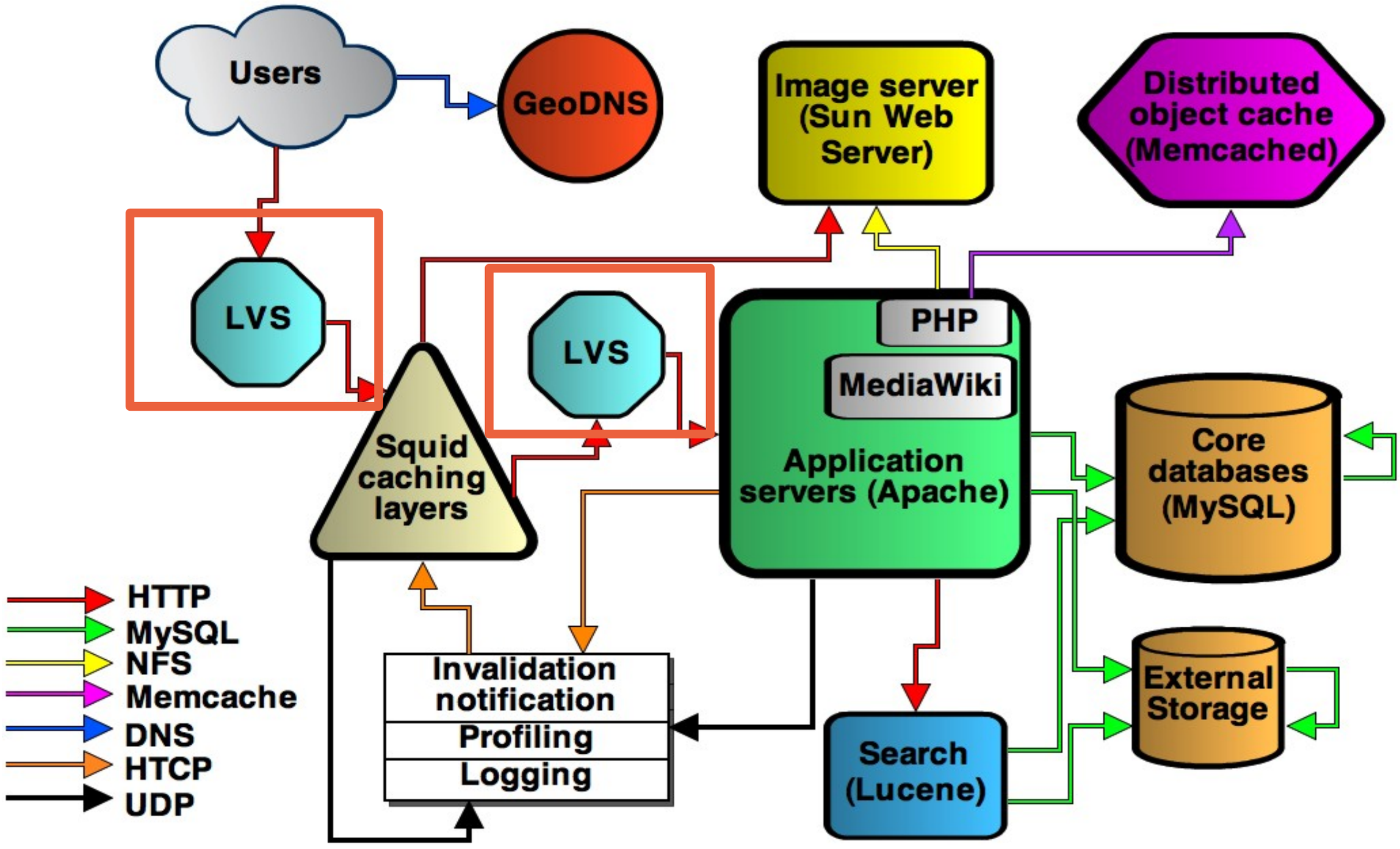
- Current solution is not scalable
- Looking at a number of solutions
- Would love help finding open source solution

Thumbnail generation

- `stat()` on each request is too expensive, so assume every file exists

- If a thumbnail doesn't exist, ask the application servers to render it



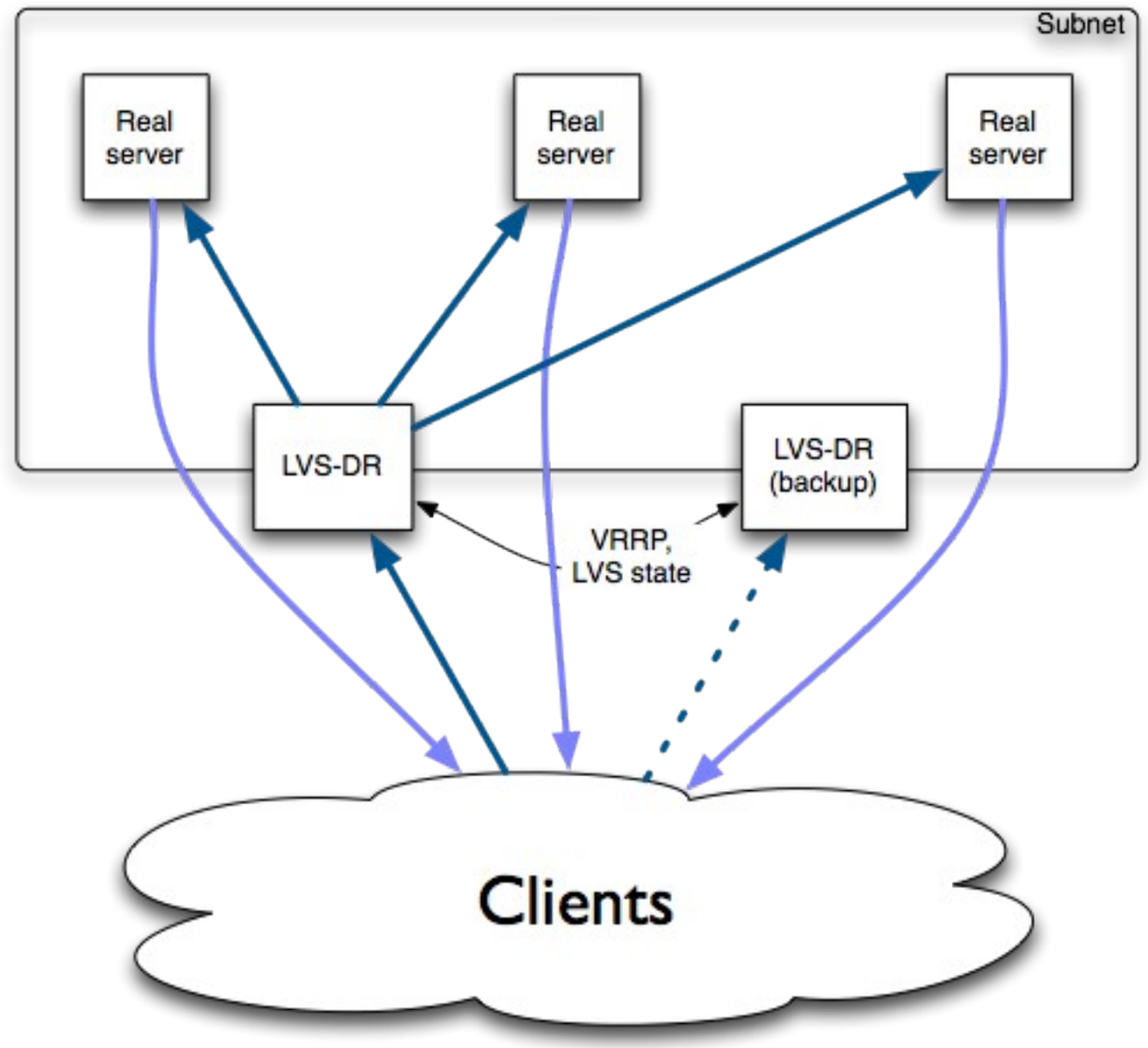


Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

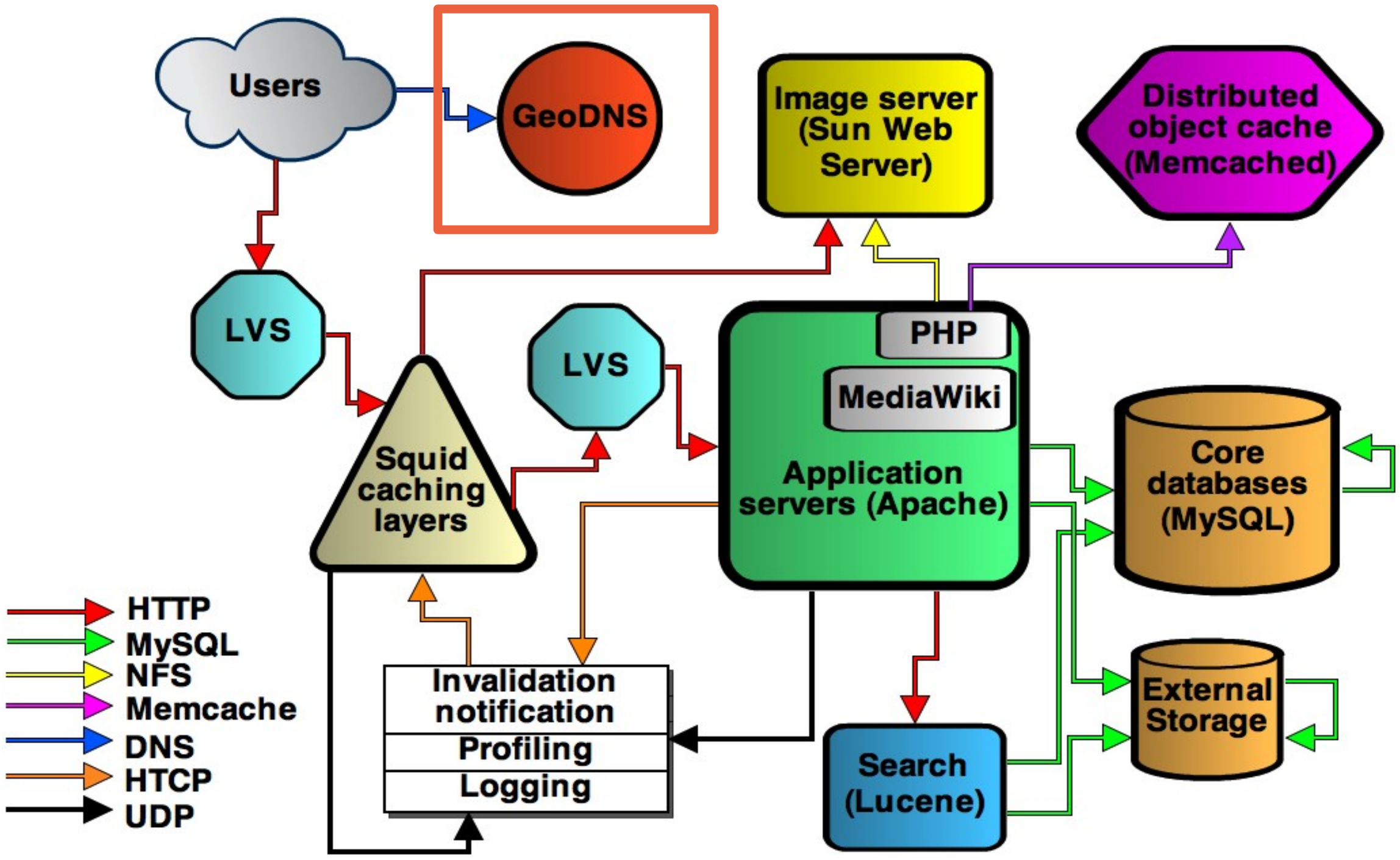
Wikimedia architecture

Load Balancing: LVS-DR

- Linux Virtual Server
- Direct Routing mode
- All real servers share the same IP address
- The load balancer divides incoming traffic over the real servers
- Return traffic goes directly!



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

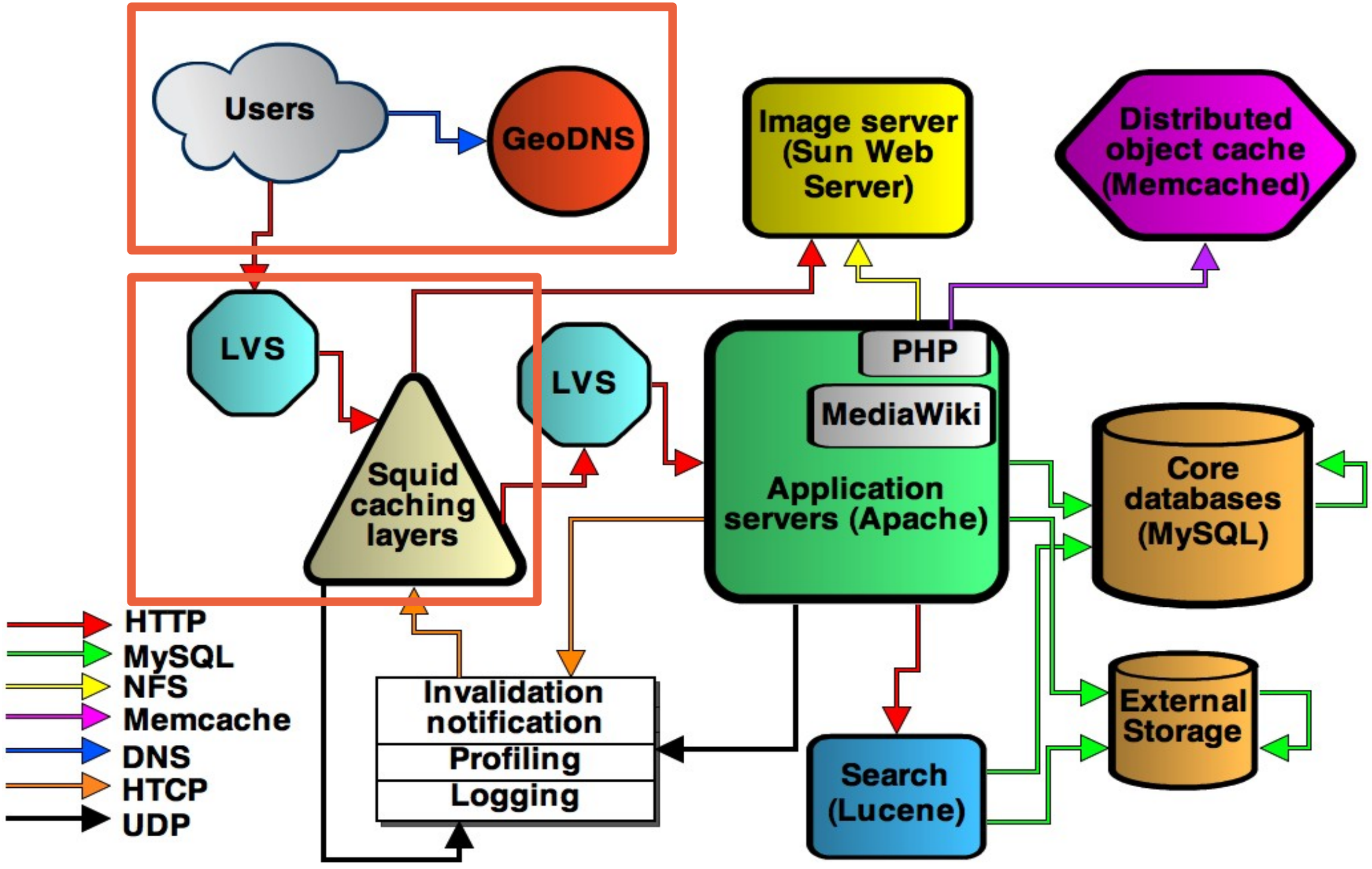


Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

Content Distribution Network (CDN)

- 2 clusters on 2 different continents:
  - Primary cluster in Tampa, Florida
  - Secondary caching-only cluster in Amsterdam
- Adding a new datacenter in Virginia, soon



Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Wikimedia architecture

Geographic Load Balancing

- Most users use DNS resolver close to them
- Map IP address of resolver to a country code
- Deliver CNAME of close datacenter entry based on country
- Using PowerDNS with a Geobackend

Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Closing notes

- We rely heavily on open source
- Always looking for efficiencies
- Looking for more efficient management tools
- Looking for more contributors

Ryan Lane, ryan@wikimedia.org, Wikimedia Foundation Inc.

Questions, comments?

- E-mail: Ryan Lane <ryan@wikimedia.org>
- IRC: Freenode, #wikimedia-tech