

Comment data stores



Node and forum pages have URLs they use to query from comment render services, and submit to comment submission services



Comment submission services receive comments against URLs and manage the comment lifecycle from submission to rendering

This might actually be one or more services all working against a message queue or event bus



Comment render services track comment counts against URLs and pointers to static rendered comment blobs



Rendered comment stores have static blobs of each comment, plus each complete page

These can be:
- memory stores that persist to disk (so there's minimal cache warmup time) like memcachedb or membase, or
- static files on real disks (not virtualized) behind an in-memory cache like Varnish (this is a little like how Boost works)



Source comment stores have unaltered comment text, IP, timestamp, user, parent, source URL



Moderation stores hold work queues, flag details, pointers to static blobs and denormalized user data



Moderation services render work queues and provide tools to edit and remove comments and discipline users

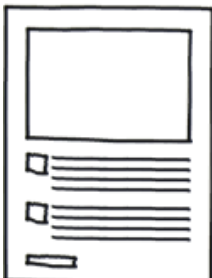


Emergency moderation services provide global comment IDs for client-side JavaScript to hide



Ignore list services are provided by Drupal as part of the friends list

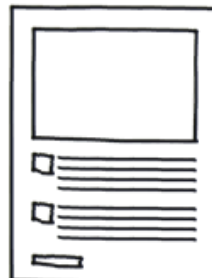
Reading a comment



Node first page



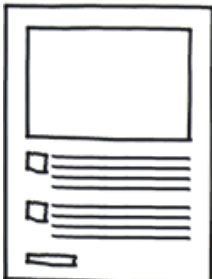
The first thirty comments are dynamic, the page includes 30 static blobs of HTML



Node page 2+



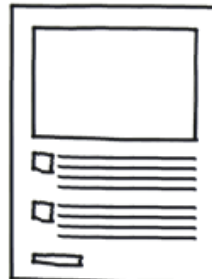
The page is including a single static blob with all 30 comments



Forum first page



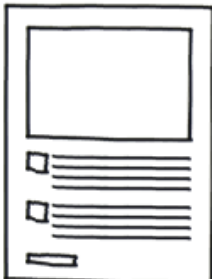
The first nine replies are dynamic, the page includes 10 static blobs of HTML



Forum first page after 10th comment



The page is a single static blob with the first post and nine replies

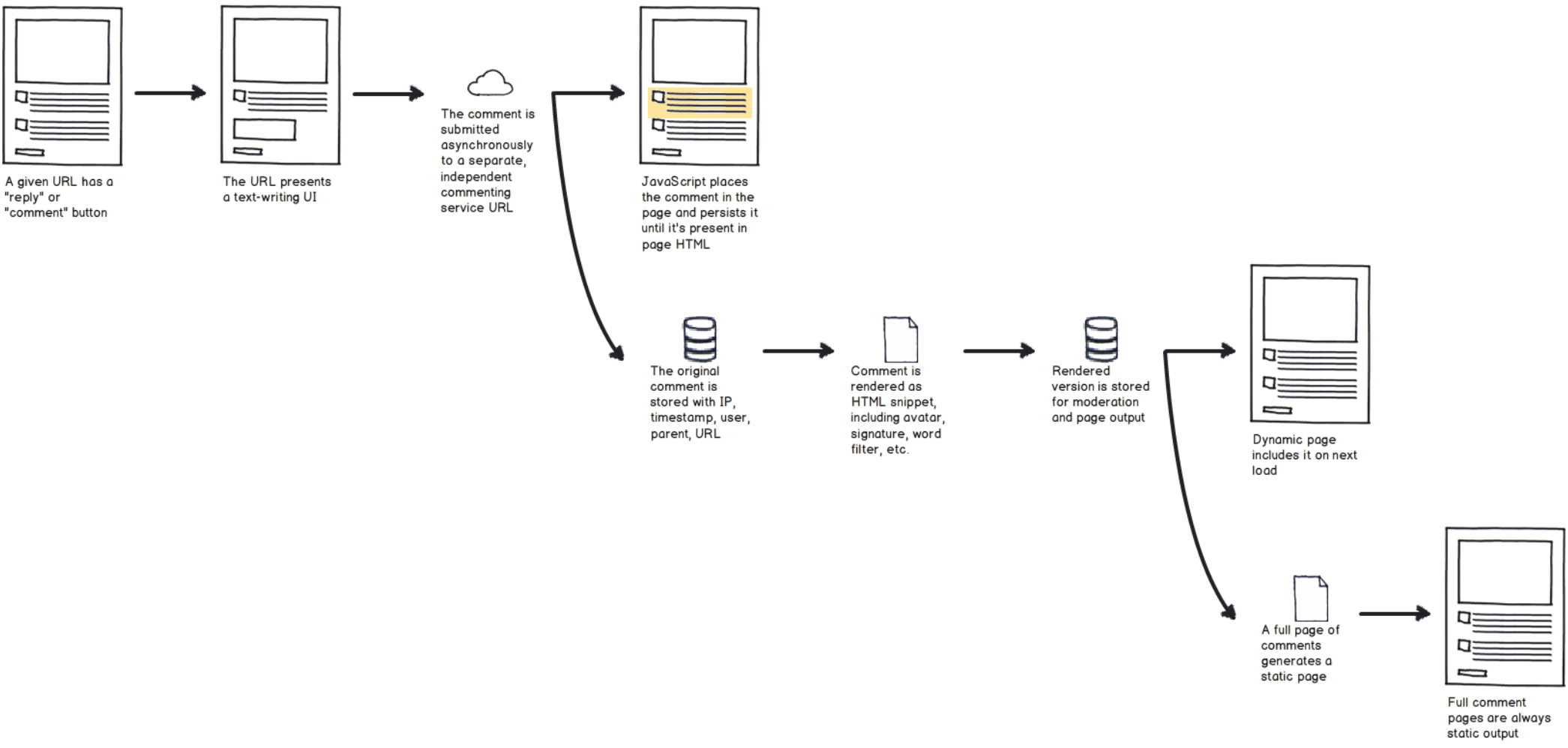


Forum page 2 after 10th comment

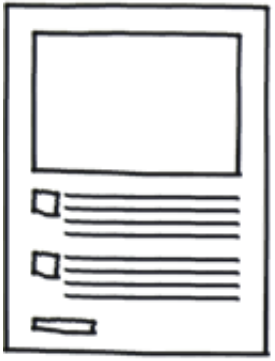


The second page is now dynamic, the page includes 10 static blobs of HTML

Submitting a comment



Flagging a comment



Users flag a post



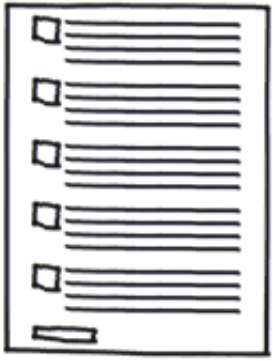
Flag information goes into moderation database



Recent user data is pulled from comment source database



Flags and denormalized user data goes into moderation queue



Moderators see static comments plus denormalized data, can request "live" data asynchronously

Moderating a comment

