



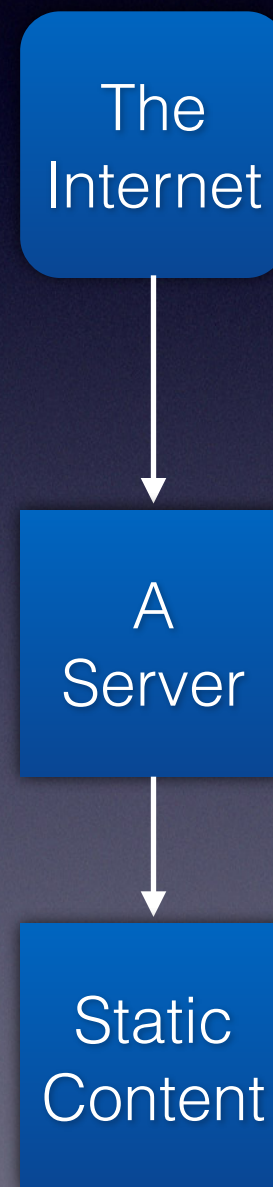
Boosting Web Application Performance

Fun with Algorithms, Architecture and Redis
@jrmiller / justinrmiller@gmail.com

A Tour of Architectures

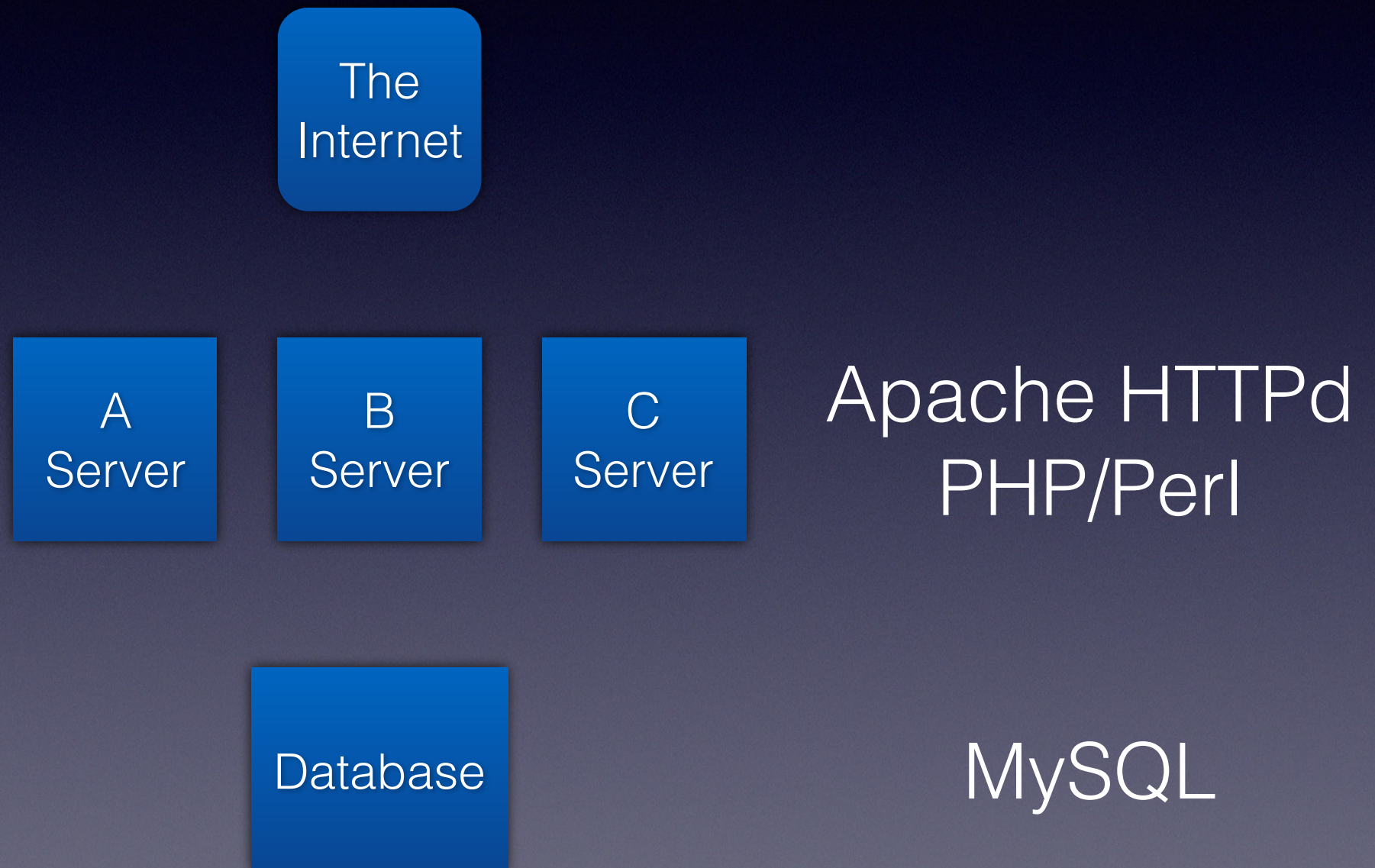
- A used christmas tree store has grown during the recession from a single store to an online superstore...

Phase 1: In the beginning...

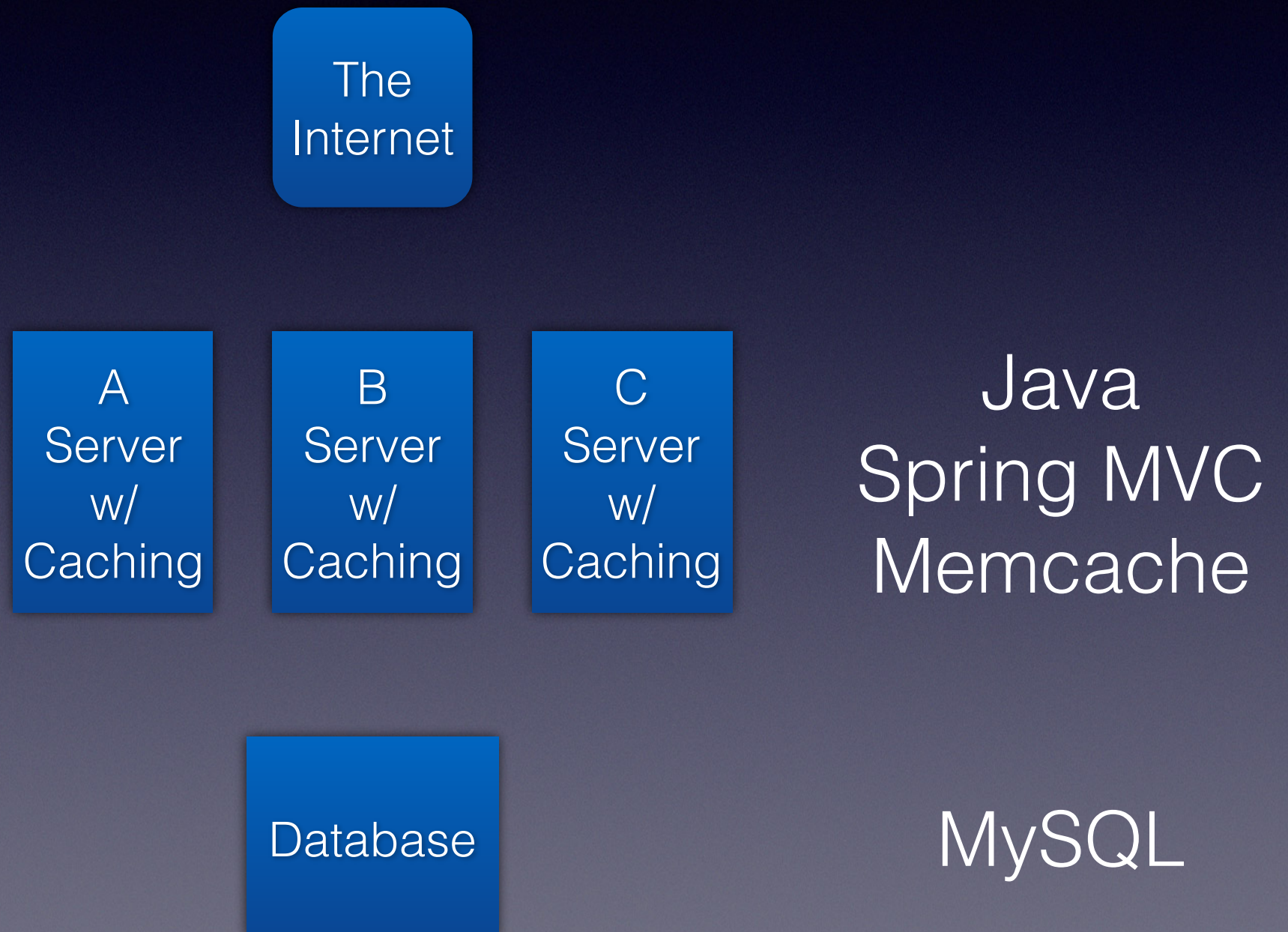


Apache HTTPd

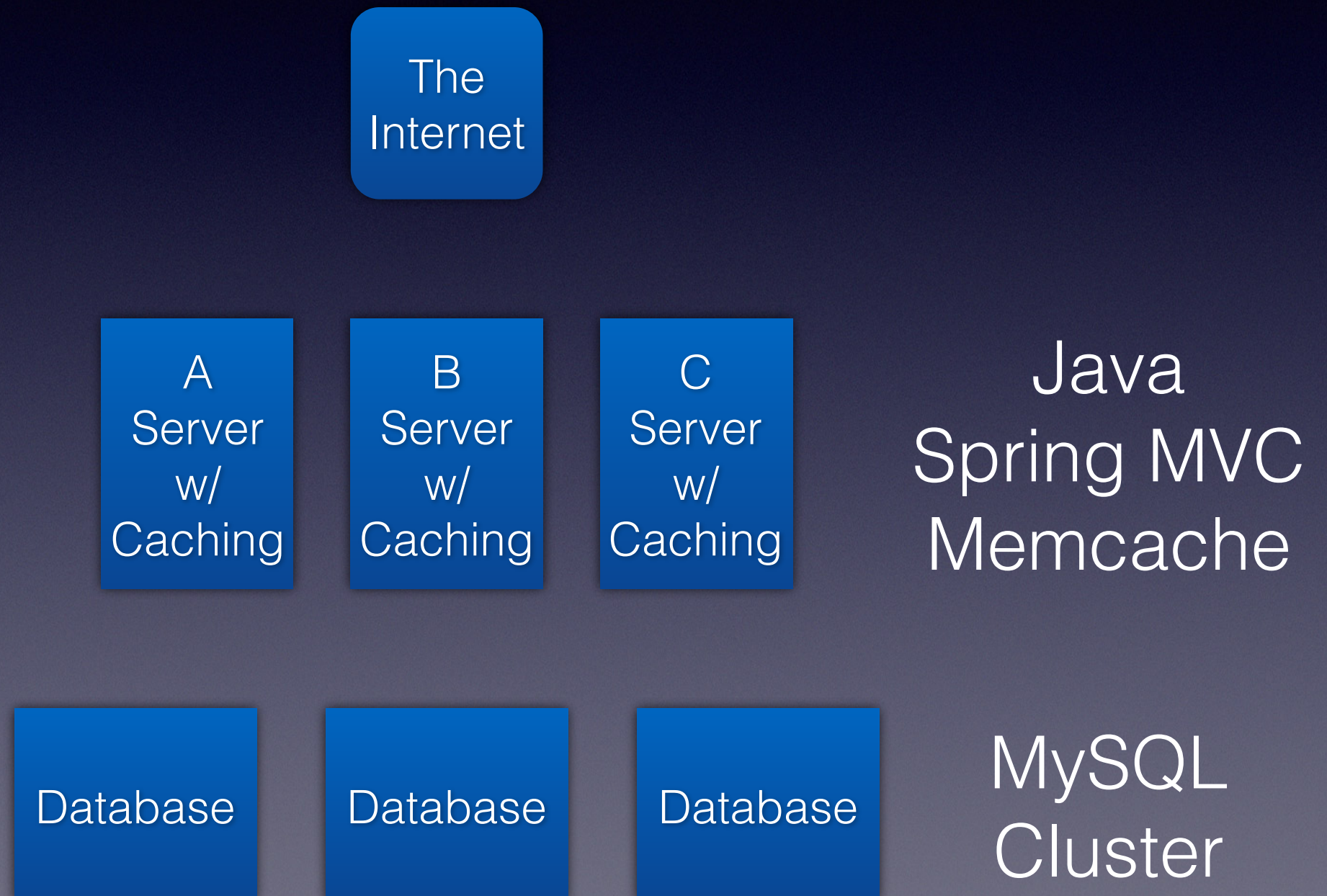
Phase 2: Shopping Cart



Phase 3: After a Long, Busy Christmas Season



Phase 4: We're rich!



Phase 5: What now?

- The site still operates in a single datacenter
 - There is a secondary data center that serves as an offsite backup and landing page for emergencies
- The pages are still rendered server side
 - 10k reqs at 10 KB/req almost saturates 1 gbit link, many pages are much larger than that
- Even with caching, the site seems sluggish

Easy Wins

- Throw all static content into a CDN
 - Product images, Audio/Video clips, Javascript
 - Faster load times around the world, fewer requests, etc.
- GZIP compression
- Re-render product images on serve to ensure you're not sending a 5 MB image for a 300x300 image
- Consider a caching HTTP reverse proxy (e.g. Varnish)

Separate Data and Presentation

- Angular - <https://angularjs.org/>
- Backbone - <http://backbonejs.org/>
- Ember - <http://emberjs.com/>
- Knockout - <http://knockoutjs.com/>
- React - <http://facebook.github.io/react/>

Why separate data and presentation?

- Front-end and Back-end can work independently (and together)
 - Requires versioned APIs or backwards compatible APIs
- Serve versioned CSS/JS/etc. from CDNs, not your own servers
- By publicly exposing your back-end services, you're forcing the back-end engineers to be super strict on validation, security, etc.

Redis

- What is it?
 - Redis is a data structure server with optional durability
 - Supports clustering (but it's beta, has been for a while)
 - Supports sets, sorted sets, hashes and more
 - Language bindings in pretty much every language

Why Redis?

- Very stable
- High performance (<http://redis.io/topics/benchmarks>)
 - SET: 552028.75 requests per second
 - GET: 707463.75 requests per second
- Fast to set up

Redis Do's

- Do
 - Ensure consistency in the way you prefix and name your keys (key utils are helpful)
 - Isolate Redis from the rest of your environment through a service/library
 - Disable Swapping (`vm.swappiness = 0`)
 - Forecast how much memory your Redis backed service will need
 - Check these tips out: <https://redislabs.com/blog/5-tips-for-running-redis-over-aws>

Redis Don'ts

- Use 64-bit Redis if you're operating on a machine with less than 4 GB of RAM ("make 32bit" will build 32-bit Redis)
- Forget to backup your Redis instance
 - Less of an issue if you don't mind your data not being there, which for caches, etc. may not be an issue
 - <http://redis.io/topics/persistence>
- Forget to setup failover if HA is your thing
 - <http://redis.io/topics/sentinel>

How to set up (Demo)

- Setting up a Redis server
 - Instructions here - <http://bit.ly/redisvagrant>

Demo Time

- All source code is available at:
 - <https://github.com/justinrmiller/spring-boot-redis-microservices>
 - mvn clean package -> generates a jar in target
 - Redis Server must be running on localhost, default redis.conf is fine
- Demos
 - Feature Flags API
 - HyperLogLog Visitor Counting API
 - Auth Token API

Redis Backed Feature Flags

- Separate the storage of feature flags from the database
- A client for the API can be written that is used by all apps which caches the feature flag data retrieved from by the API from Redis
- See FeatureFlagsController

HyperLogLog Unique Visitor Count

- <http://en.wikipedia.org/wiki/HyperLogLog>
- “The HyperLogLog algorithm is able to estimate cardinalities of $>10^9$ with a typical accuracy of 2%, using 1.5kB of memory.”
- Translation: Store huge unique visitor counts with very little in the way of time/space requirements
- See HyperLogLogController

Auth Token API

- Provide a facade for services hosted internally
- An auth token provides protection against abuse of services by hackers and customers
- See AuthTokenController

Questions?

- Slides and source code are available at:
 - <http://bit.ly/redispresentation>
- Cool links
 - <https://cloud.google.com/solutions/redis/click-to-deploy>
 - <http://instagram-engineering.tumblr.com/post/12202313862/storing-hundreds-of-millions-of-simple-key-value>
 - <http://engineering.sharethrough.com/blog/2013/06/06/how-redis-took-us-offline-and-what-we-did-about-it/>