

Building a search platform 101

Anshum Gupta

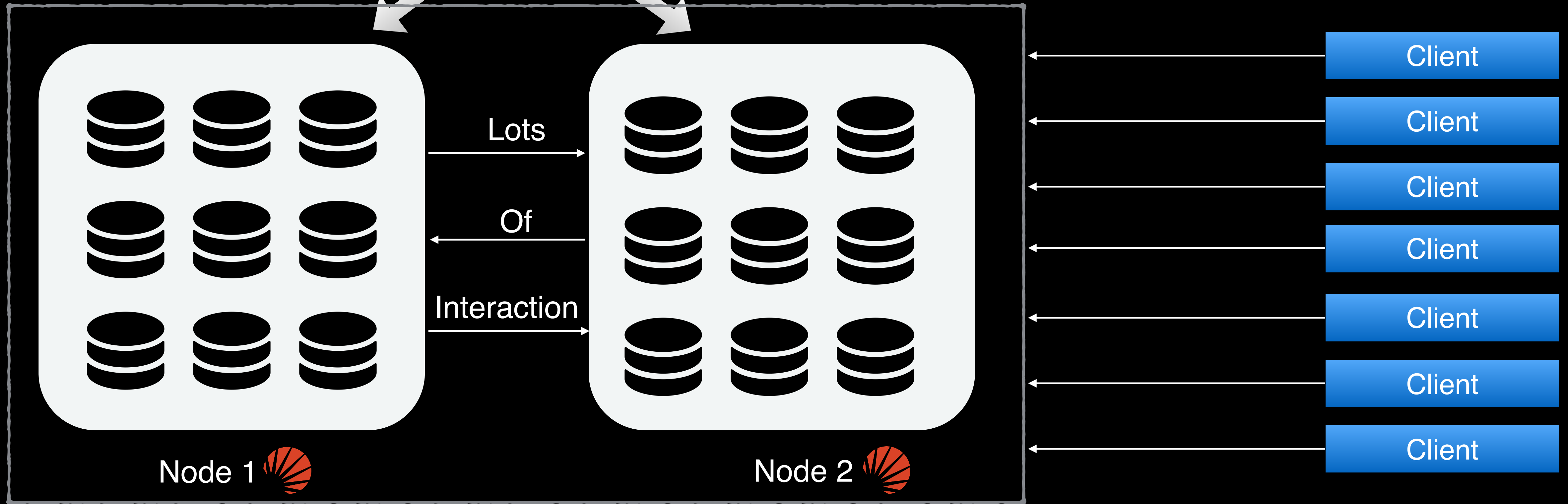
Apache Lucene/Solr Committer, and PMC member



A bit about myself

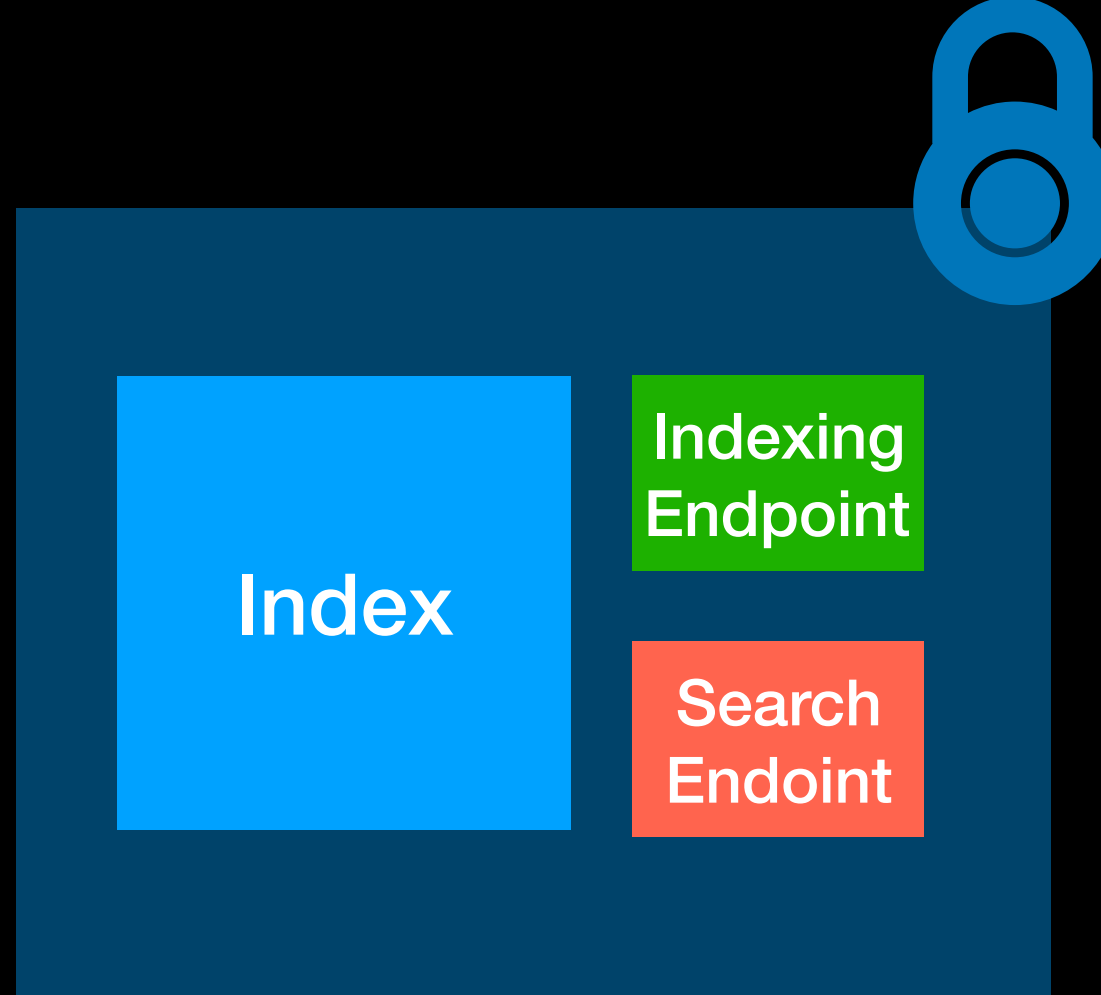
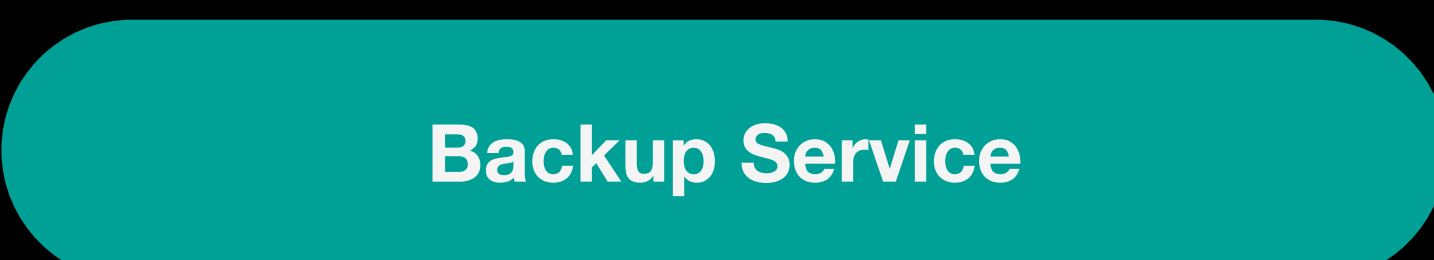
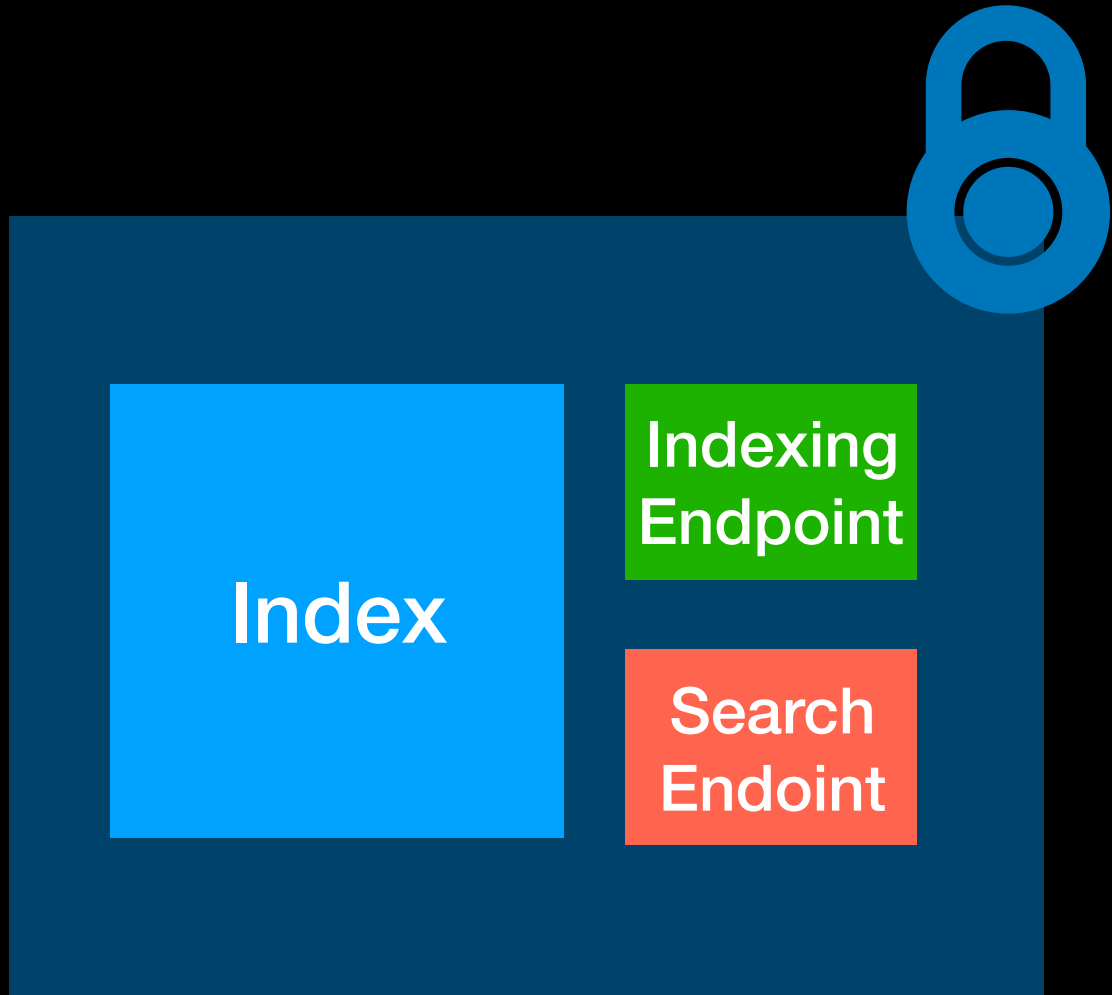
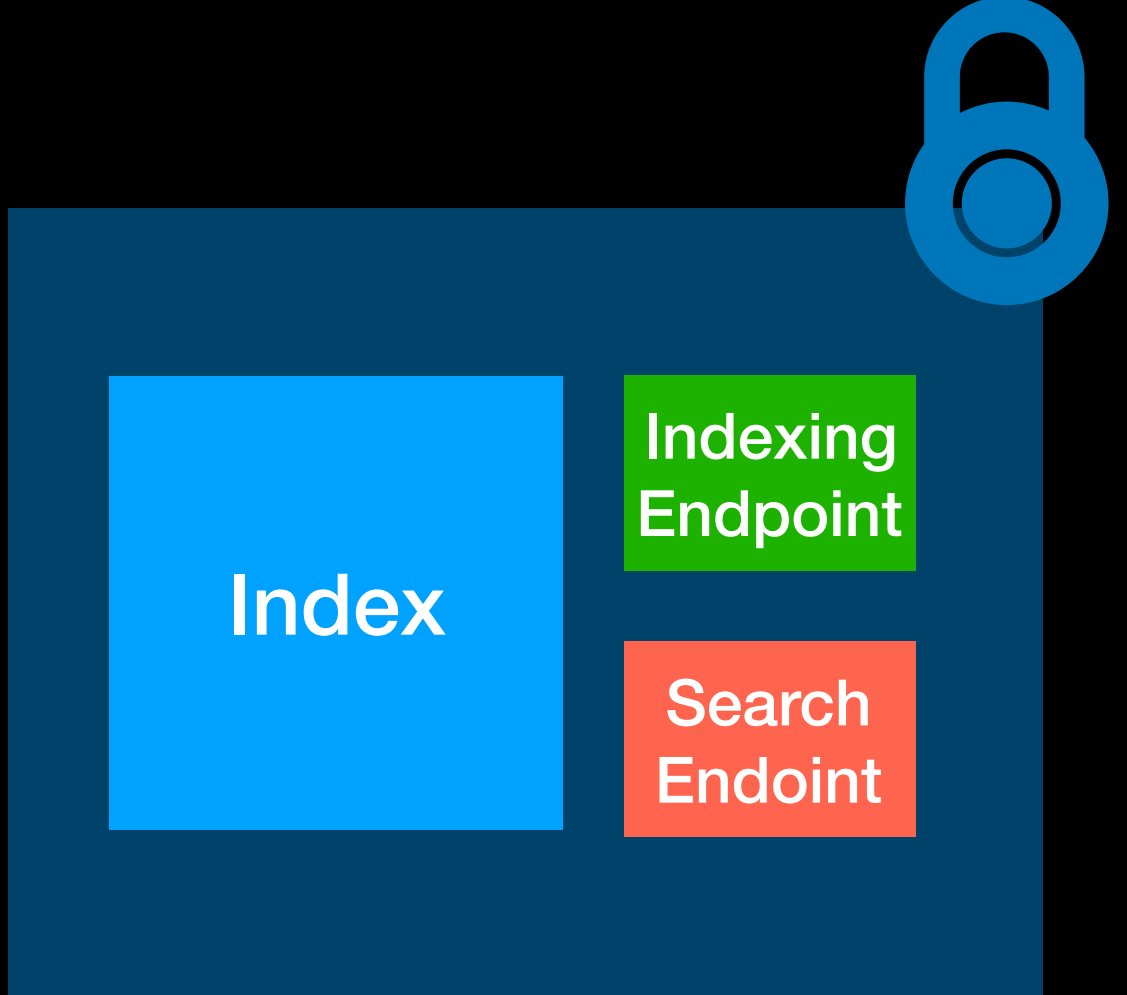
- Love “Distributed” Search
- Apache Lucene/Solr since 2006
- Built search platforms at a lot of organizations
- But more importantly, Open Source @ Apache Lucene / Solr :)
- Committer and, PMC member for a while now

Search Engine... as we know it



Coins by Creative Stall from the Noun Project


Search Platform



Search Platform

**_Why_ do you need a
platform?**

I think you already know

- Consolidate non-trivial effort
 - Make it easier for users to setup search
 - Make it cost-effective due to
 - Almost trivially repeatable tasks
- 

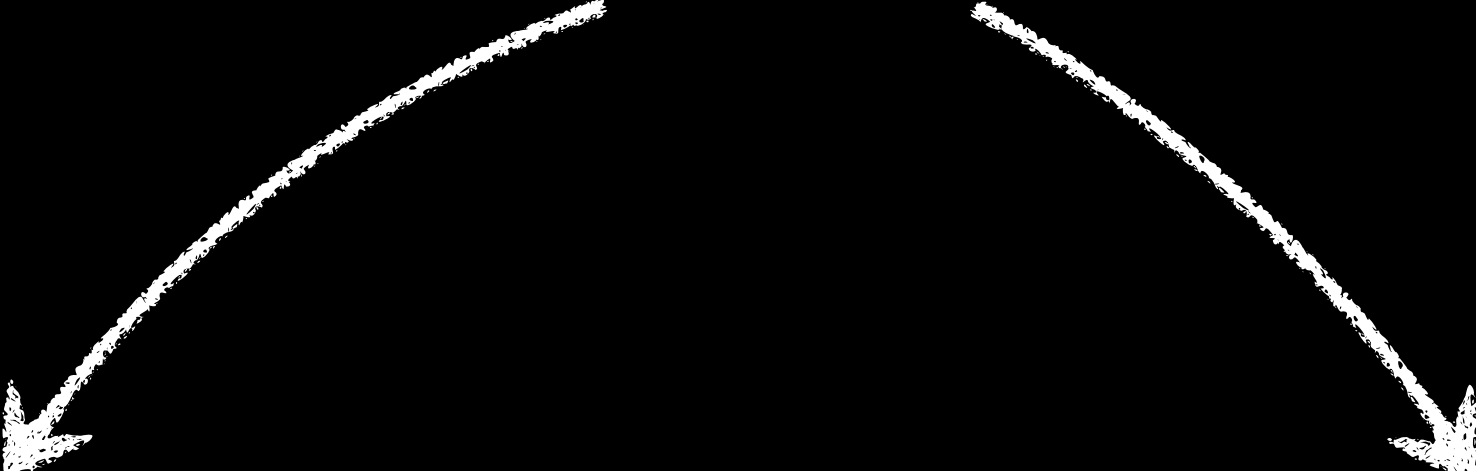
**Before you begin, ask these
questions!**

Who is the user?



It's most important to understand the user!

Broad Classification



Internal

External

More liberal

Strict and Guarded

Exposure

Complexity

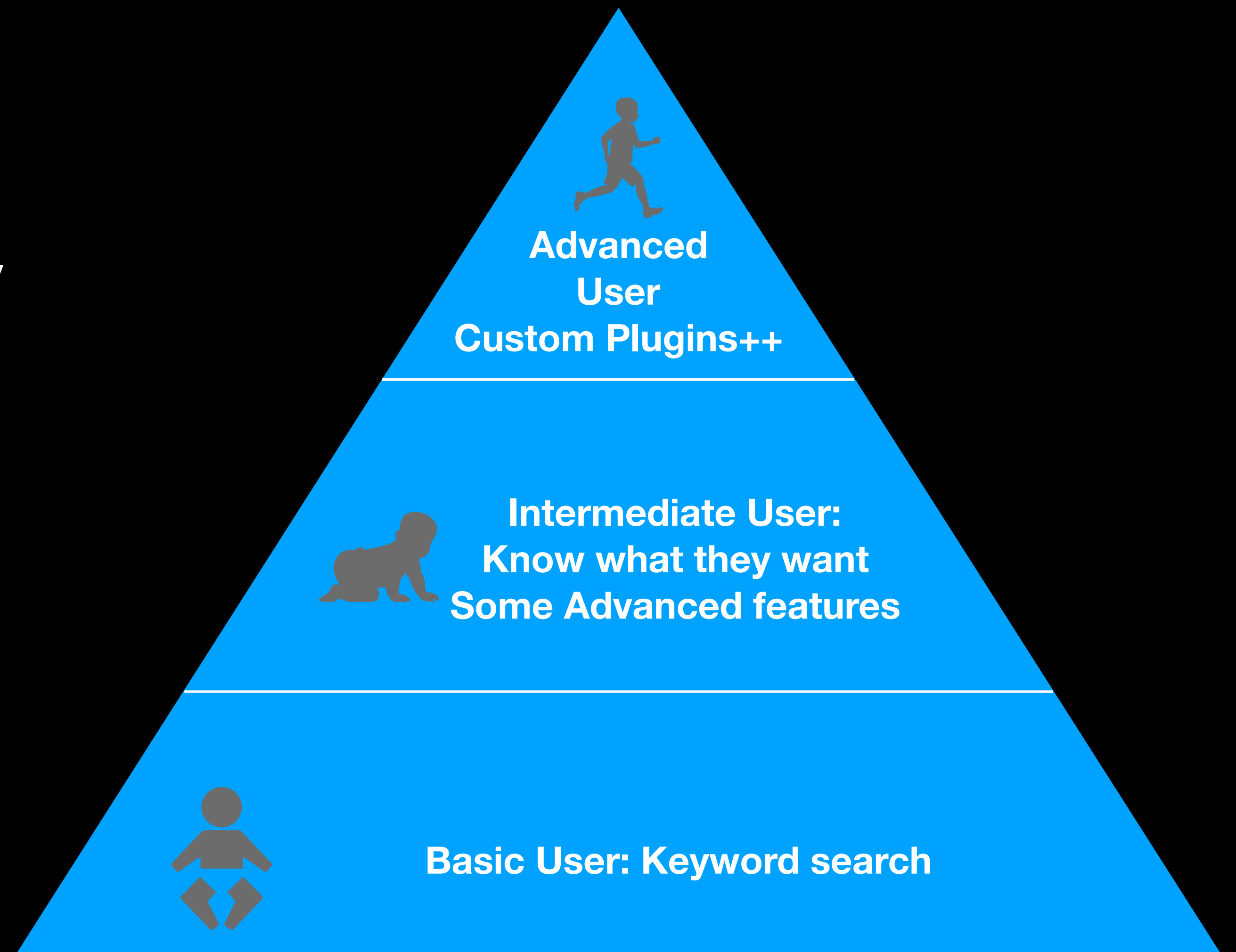
Access

Logging

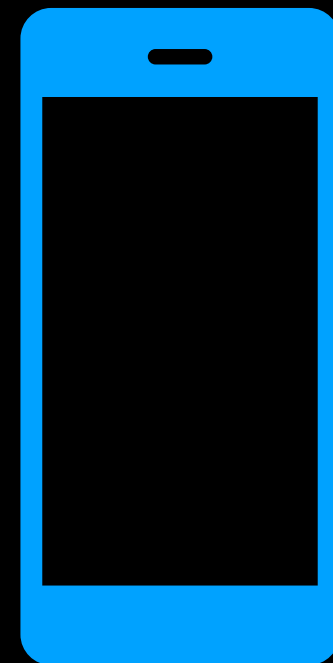
Shared Resources

Where in this pyramid is the user ?

- Three basic categories of users:
 - Basic
 - Auto-detection of fields (please, don't try this for anyone else)
 - Intermediate - Most challenging
 - Custom analyzer chains, faceting
 - Advanced
- Helps in deciding which features to expose
- Based on the answer to the previous question, concentrate on either top OR bottom of the pyramid to begin with



**How will the service
interaction happen?**



- Repeat to yourself - “Customer APIs can not change”
- Spend a lot of thought in designing
 - Endpoint
 - APIs - Client, and REST
- Compatibility in the future - think about the APIs

What's generally needed

- Safeguarding - Restriction of requests
- Abstraction of ZooKeeper - Customers should not know about it
- Throttling
- Fail fast for unsupported features
- Stable APIs!

SolrClient - Pros

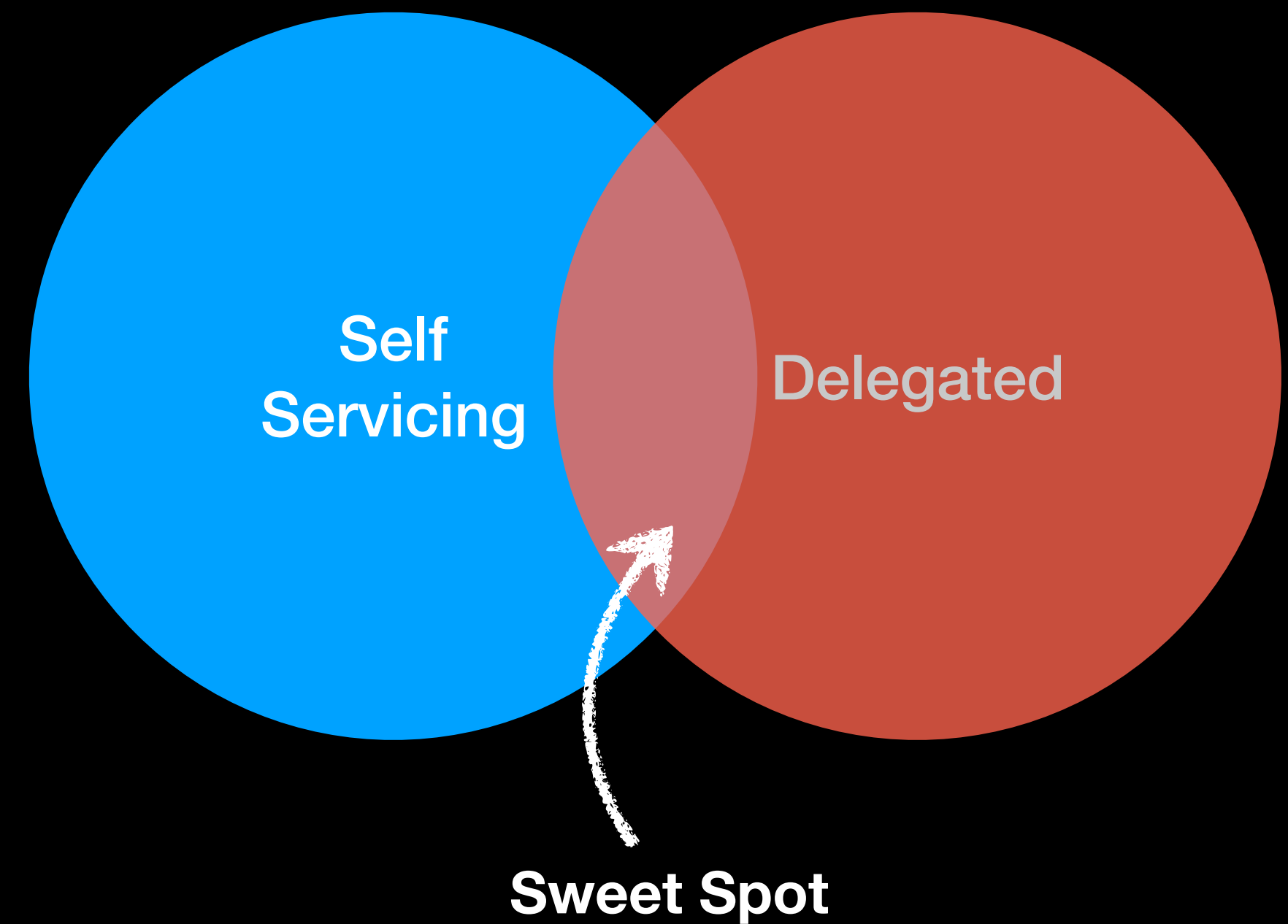
- Piggyback on Solr APIs
 - No need to document everything
 - Easy to add custom features

But why not ?

- CustomClient : SolrClient :: Search Platform: Vanilla Solr
- Exposes ZooKeeper
- Need to restrict features
- Experimental APIs
- A LOT of flux

**Who will manage the
clusters?**

- Delegated
 - Requires auto-scaling
 - Not supported out of the box in Solr completely - work in progress
 - Requires restriction, OR magic
- Self servicing
 - User managed
 - Idiot proofing needed - easier to shoot themselves in the foot BUT you are liable!
 - Fire fight when beyond control - no alarms raised in time
- Combination of the two
 - Basic auto-scaling - tiered
 - Allow them to go wide, spin up new replicas, and more!



How will you `_monitor_`?



X-Ray Vision!

- How much is exposed to the end user
 - Admin interface
 - Not everything should be exposed*
- Monitor metrics as much as possible
 - JMX - gives you snapshot, so you need a plan to hold this information somewhere.
 - Newer versions of Solr have metrics reporters, you might want to use those.

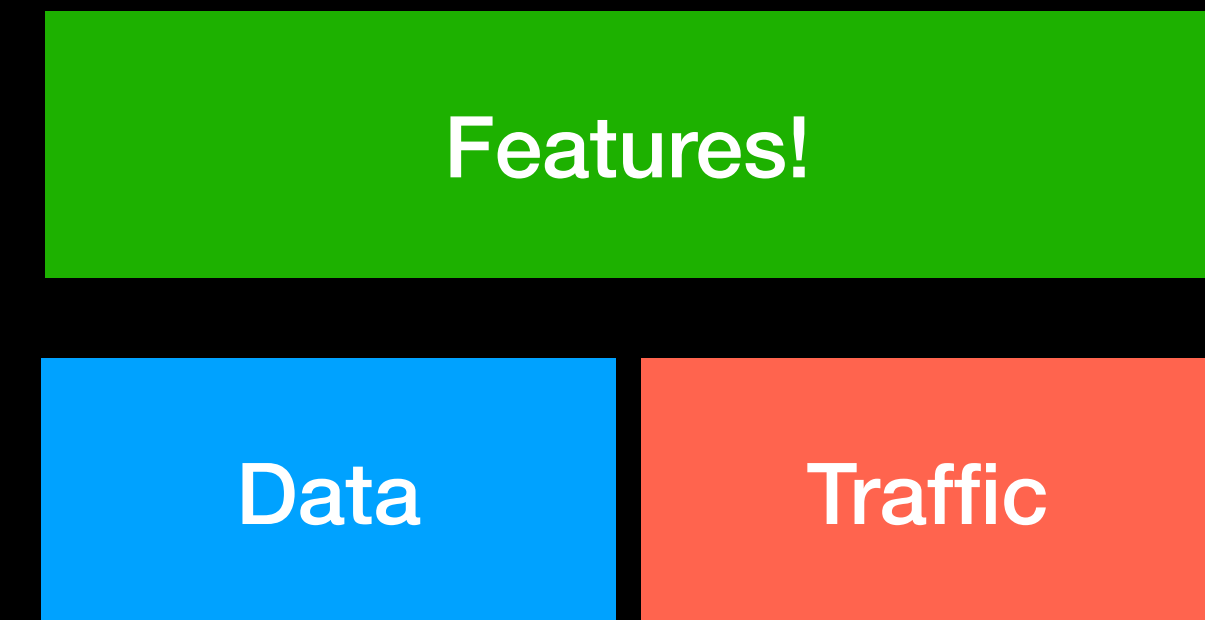
* unless the platform is only used by advanced users

What's the plan to `_scale_`?

Start small, and grow gradually!

Scalability - Now and the future

- Data
 - Begin with limited allowance
 - Always remember, this will grow
- Traffic - set limits
 - No back-pressure in Solr, remember that.
 - Graceful degradation
- Most important: Features
 - Not all features are equal!



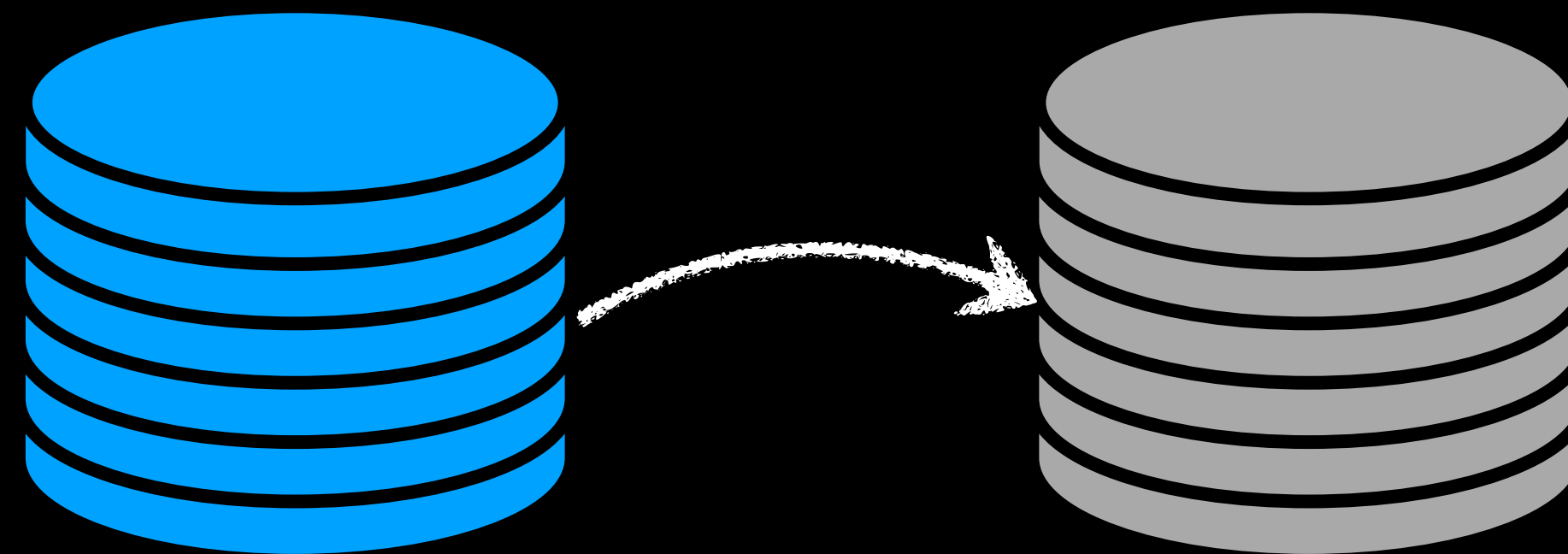
How will you manage versioning, and `_upgrades_` ?

As much as you may try, there will always be more than one versions

- Almost always need to support multiple versions in the long run
- Ideally, there should be only one production version
- Proxy - Intermediate layer, that is transparent of the back-end version
- User facing API doesn't change = Upgrade at will, and roll back too*!

*** Not that you should need to roll back often**

What about `_backups_` ?



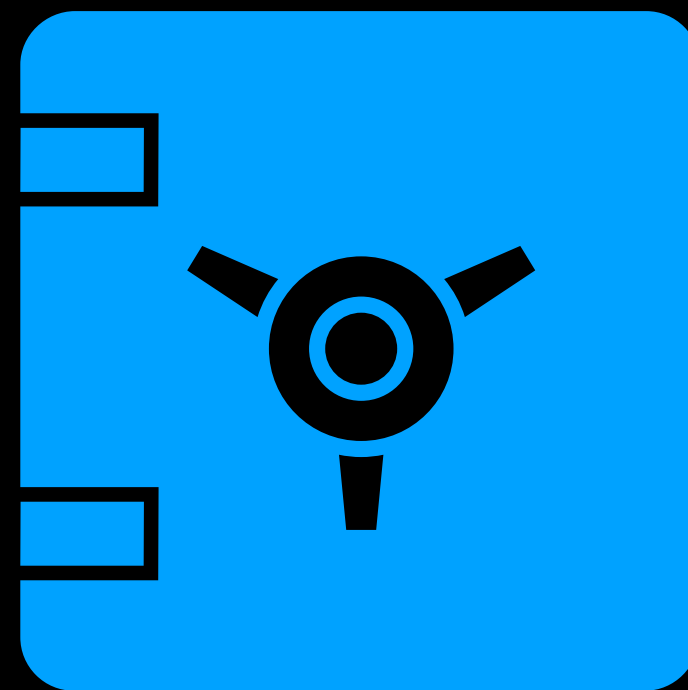
Because things don't always go a planned

- How often do you want to backup?
- Backups in SolrCloud are expensive
- Only recent versions support it
- Plan around cleaning up of backups - you can't hold them forever on live disks

**Do you want to support auto
reindex ?**

- Would your users need to reindex data often?
 - If so, it might be worth storing it all
 - Even if you do, DO NOT use it as the primary store!
- Consider using an alias, the client code wouldn't need to change
- Proxy is a good place to manage such things!

What are your plans about cluster_security_?

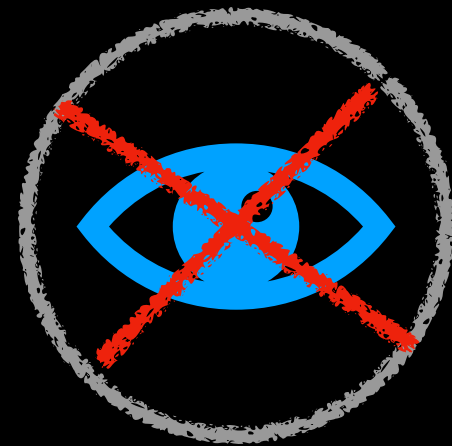


Security over everything, well, almost always!

- It's complicated :/
- What kind of security is required?
 - Take that answer and bump it up a bit - that is how much you'd need.
- Don't forget about the satellite systems!
- Does Solr support the security mechanism? Think auth plugins
 - Solr manages intra-cluster/inter-node communication

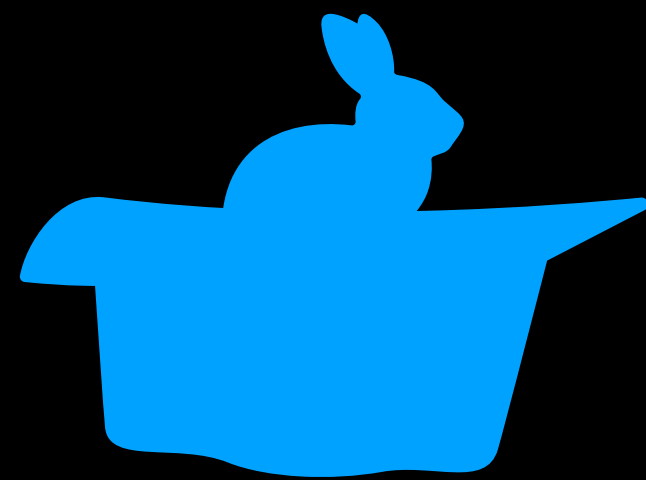
Authentication	Authorization	Proxy - Cluster Security
SSL	ZK - Solr w/ SSL	Encryption of Logs/Backups
Kerberos	Document / Field Level	On-Disk Encryption

Is the indexed data
sensitive ?



- Can you log requests
 - If not - find a way to obfuscate requests and track
- Do you have access to the data ?

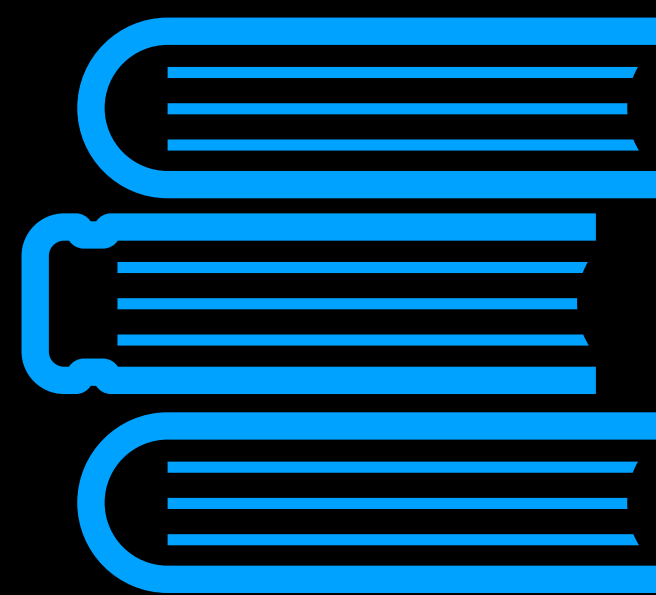
Do you or your customers have
custom plugins ?



Magic Tricks!

- Everyone feels that their plugin is ‘essential’, but is it?
 - Analyzers, rankers, and more
- It’s hard to maintain when the plugin writer isn’t the platform owner
- Standardize plugins, or do it outside
- If it’s an advanced platform
 - Blob store: Solr has a mechanism to distribute plugins using API

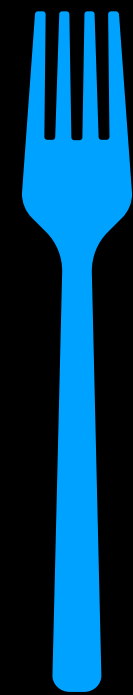
How will you `_train_` users?



Library? :)

- Often overlooked question
- Hand holding is hard
- Demos, and examples
- Not just getting started but also
 - APIs,
 - Code examples
- Platform to teach about best practices

One more thing! Do you plan to
fork?

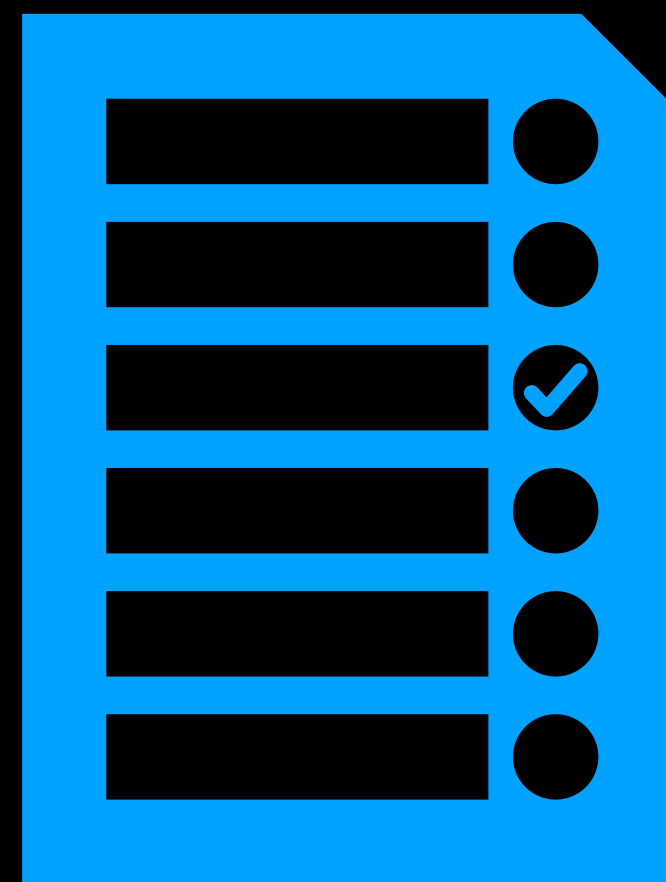


No! Please Don't!*



***if you do so, please read the offer document carefully before investing.**

Summary



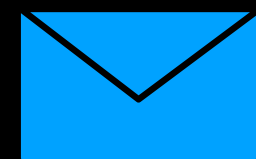
- Keep it simple
- Only cater to a few use cases to begin with
- Don't bite off more than you can chew
- Start with the most recent stable release and don't try to keep up with the releases - not always.



<http://www.twitter.com/anshumgupta>



<http://www.linkedin.com/in/anshumgupta/>



anshum@apache.org