Rebuilding Slack Search with Armeria

November 21, 2019

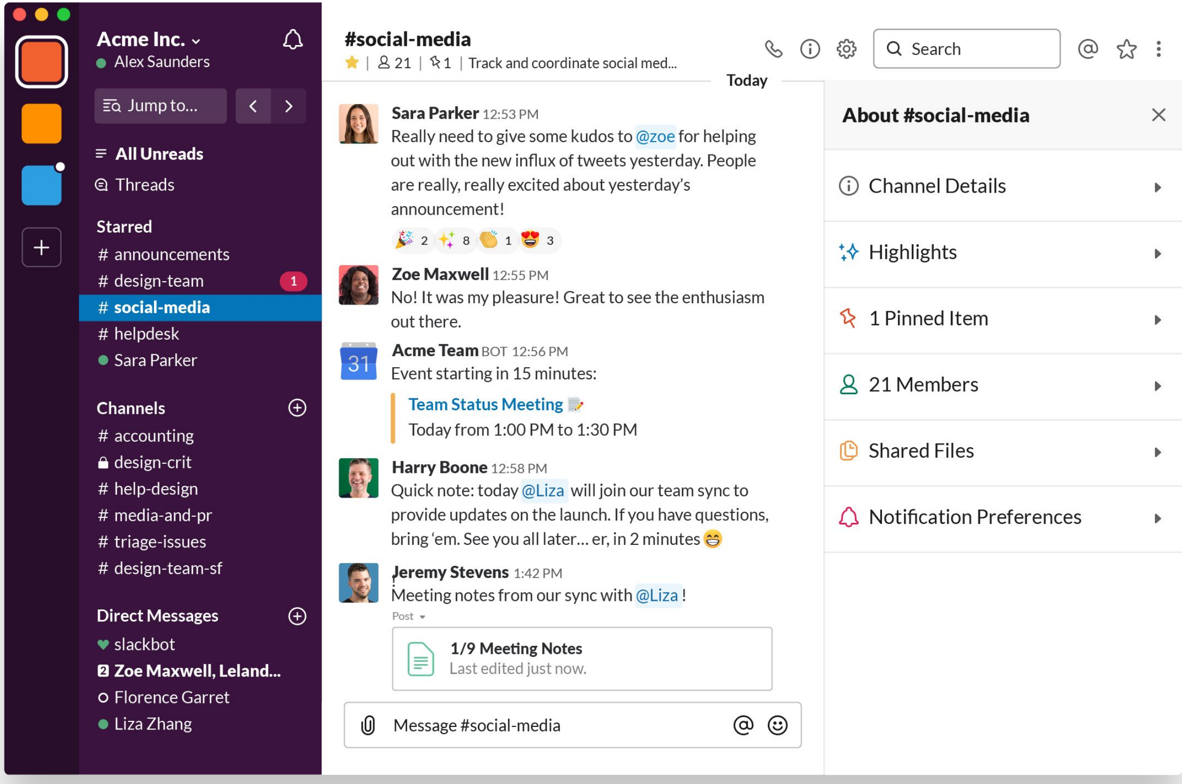
slack

Renaud Bourassa

Staff Software Engineer
Search & Discovery

Slack

Our mission is to make people's working lives simpler, more pleasant, and more productive.



Sara Parker 12:53 PM
Really need to give some kudos to @zoe for helping out with the new influx of tweets yesterday. People are really, really excited about yesterday's announcement!
🎉 2 ✨ 8 👏 1 😍 3

Zoe Maxwell 12:55 PM
No! It was my pleasure! Great to see the enthusiasm out there.

Acme Team BOT 12:56 PM
Event starting in 15 minutes:
Team Status Meeting 📝
Today from 1:00 PM to 1:30 PM

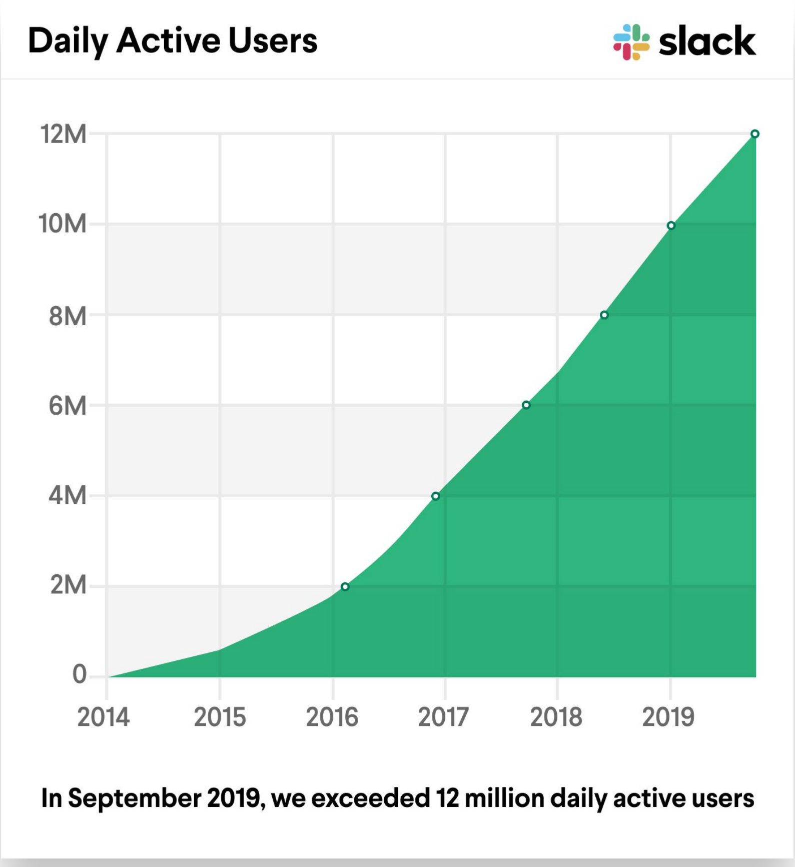
Harry Boone 12:58 PM
Quick note: today @Liza will join our team sync to provide updates on the launch. If you have questions, bring 'em. See you all later... er, in 2 minutes 😁

Jeremy Stevens 1:42 PM
Meeting notes from our sync with @Liza !
Post
1/9 Meeting Notes
Last edited just now.

Stats

12M+ daily active users (half outside of North America)

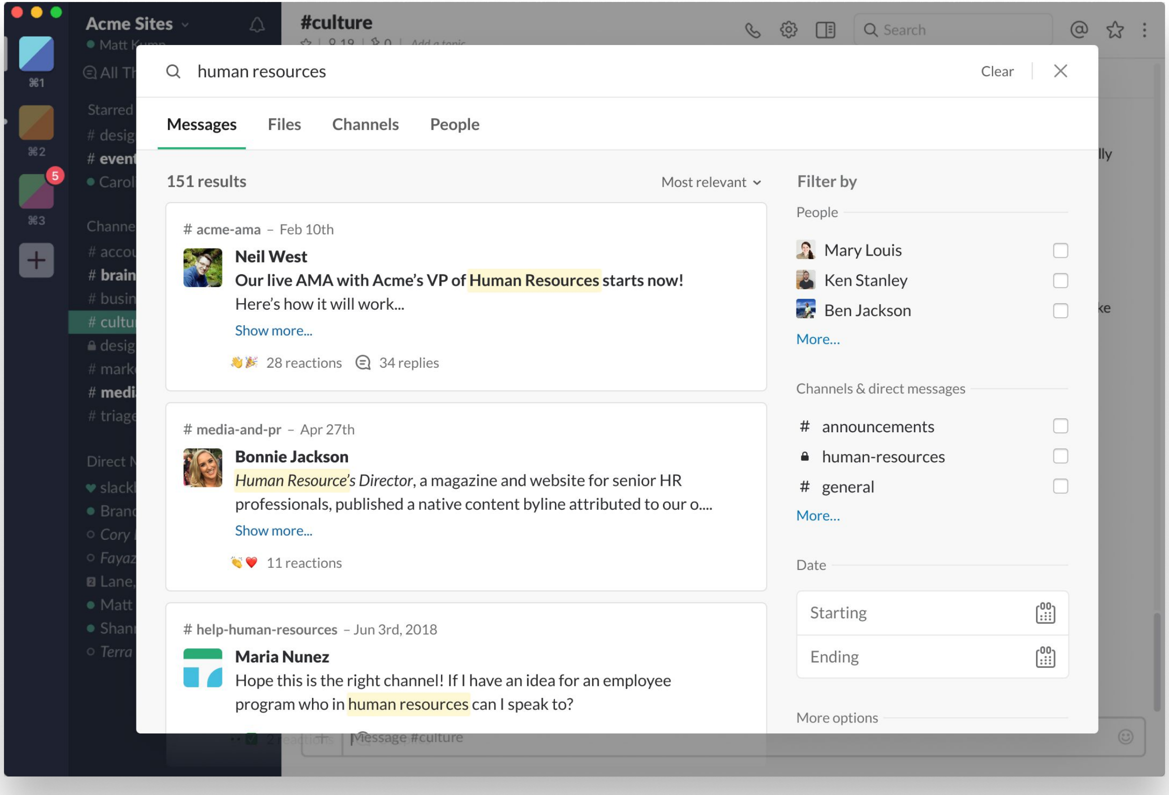
5B+ actions/week



In September 2019, we exceeded 12 million daily active users

Search & Discovery

We provide tools to help people find, discover, organize and focus on the information they need to get work done in Slack.



human resources

151 results

# acme-ama – Feb 10th

Neil West
Our live AMA with Acme's VP of Human Resources starts now!
Here's how it will work...
Show more...
28 reactions 34 replies

# media-and-pr – Apr 27th

Bonnie Jackson
Human Resource's Director, a magazine and website for senior HR professionals, published a native content byline attributed to our o....
Show more...
11 reactions

# help-human-resources – Jun 3rd, 2018

Maria Nunez
Hope this is the right channel! If I have an idea for an employee program who in human resources can I speak to?

People
- Mary Louis □
- Ken Stanley □
- Ben Jackson □

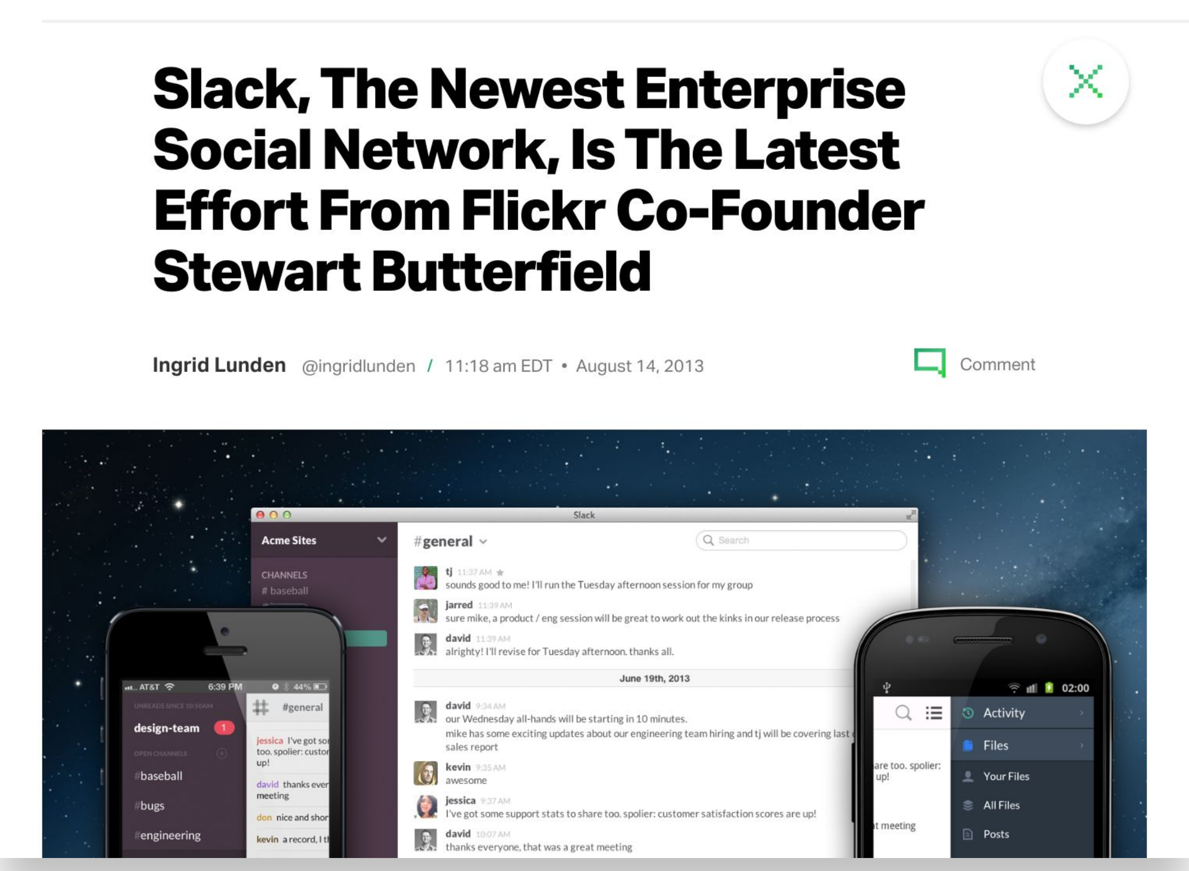
Channels & direct messages
- # announcements □
- human-resources □
- # general □

Date
- Starting
- Ending

A Brief History of Search

August 2013

Slack open beta launches with Search as a core feature.



Slack, The Newest Enterprise Social Network, Is The Latest Effort From Flickr Co-Founder Stewart Butterfield

Ingrid Lunden @ingridlunden / 11:18 am EDT • August 14, 2013

Comment

#general

tj 11:37 AM
sounds good to me! I'll run the Tuesday afternoon session for my group

jarred 11:39 AM
sure mike, a product / eng session will be great to work out the kinks in our release process

david 11:39 AM
alrighty! I'll revise for Tuesday afternoon. thanks all.

June 19th, 2013

david 9:34 AM
our Wednesday all-hands will be starting in 10 minutes.
mike has some exciting updates about our engineering team hiring and tj will be covering last
sales report

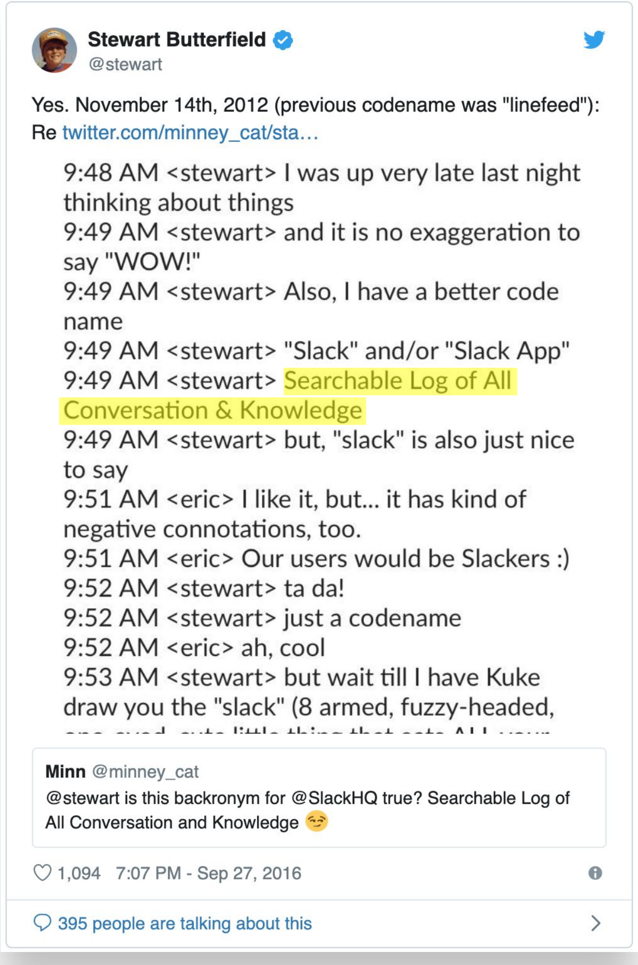
kevin 9:35 AM
awesome

jessica 9:37 AM
I've got some support stats to share too. spolier: customer satisfaction scores are up!

david 10:07 AM
thanks everyone, that was a great meeting

November 2012

Search was always seen as core to the Slack experience.



Stewart Butterfield @stewart

Yes. November 14th, 2012 (previous codename was "linefeed"): Re twitter.com/minney_cat/sta…

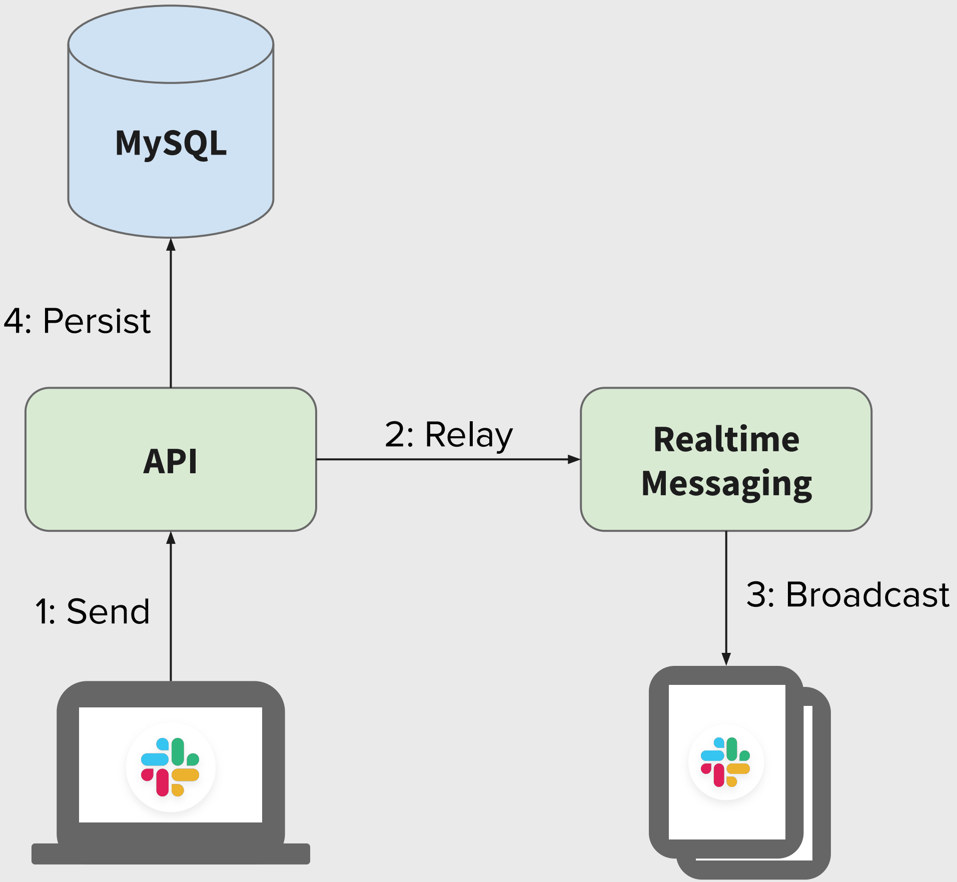
9:48 AM <stewart> I was up very late last night thinking about things
9:49 AM <stewart> and it is no exaggeration to say "WOW!"
9:49 AM <stewart> Also, I have a better code name
9:49 AM <stewart> "Slack" and/or "Slack App"
9:49 AM <stewart> Searchable Log of All Conversation & Knowledge
9:49 AM <stewart> but, "slack" is also just nice to say
9:51 AM <eric> I like it, but... it has kind of negative connotations, too.
9:51 AM <eric> Our users would be Slackers :)
9:52 AM <stewart> ta da!
9:52 AM <stewart> just a codename
9:52 AM <eric> ah, cool
9:53 AM <stewart> but wait till I have Kuke draw you the "slack" (8 armed, fuzzy-headed, one eyed, cute little thing that eats ALL your

Minn @minney_cat
@stewart is this backronym for @SlackHQ true? Searchable Log of All Conversation and Knowledge 😏

♡ 1,094 7:07 PM - Sep 27, 2016

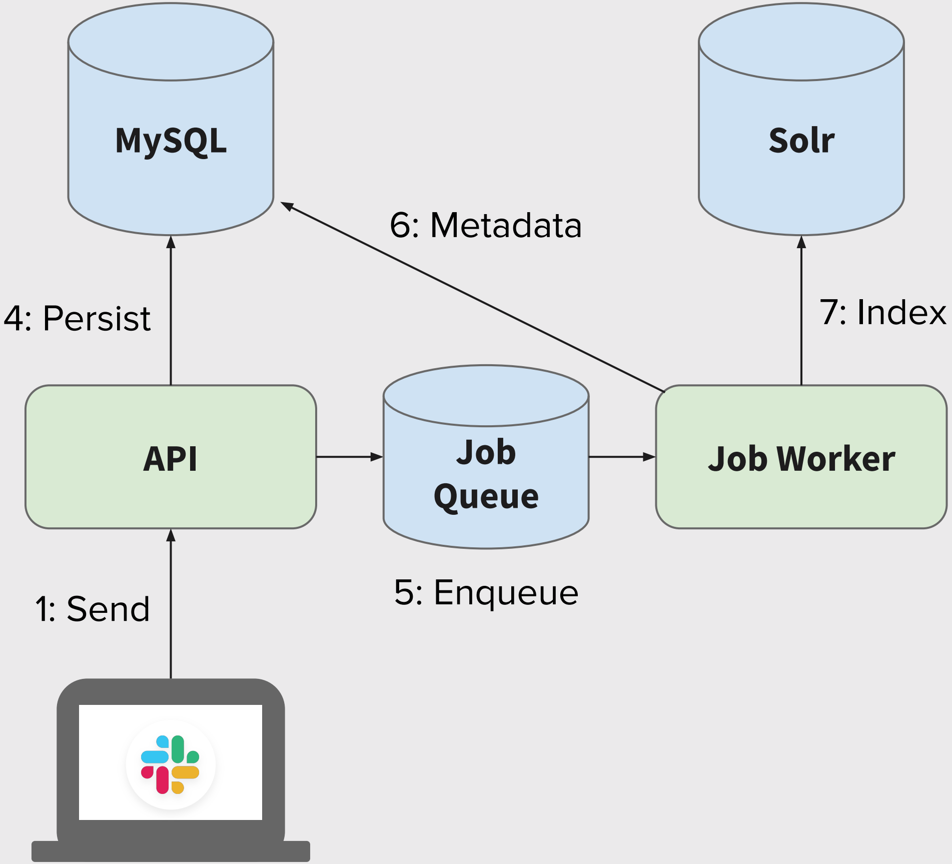
395 people are talking about this

Slack Architecture



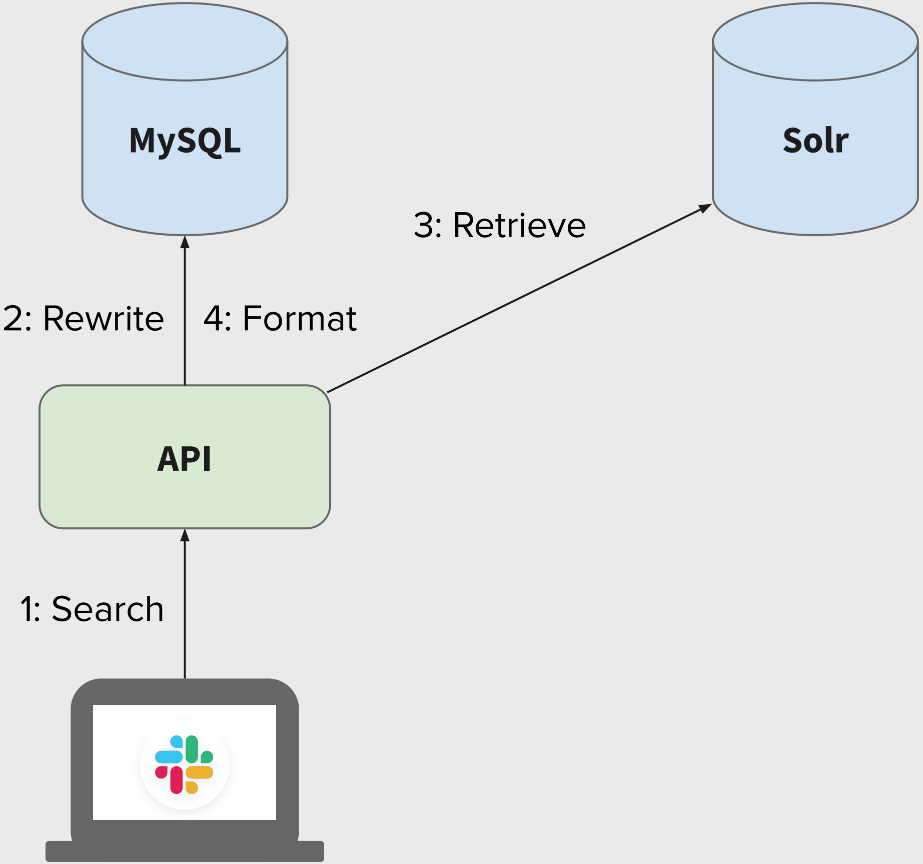
Search Indexing

Write Path



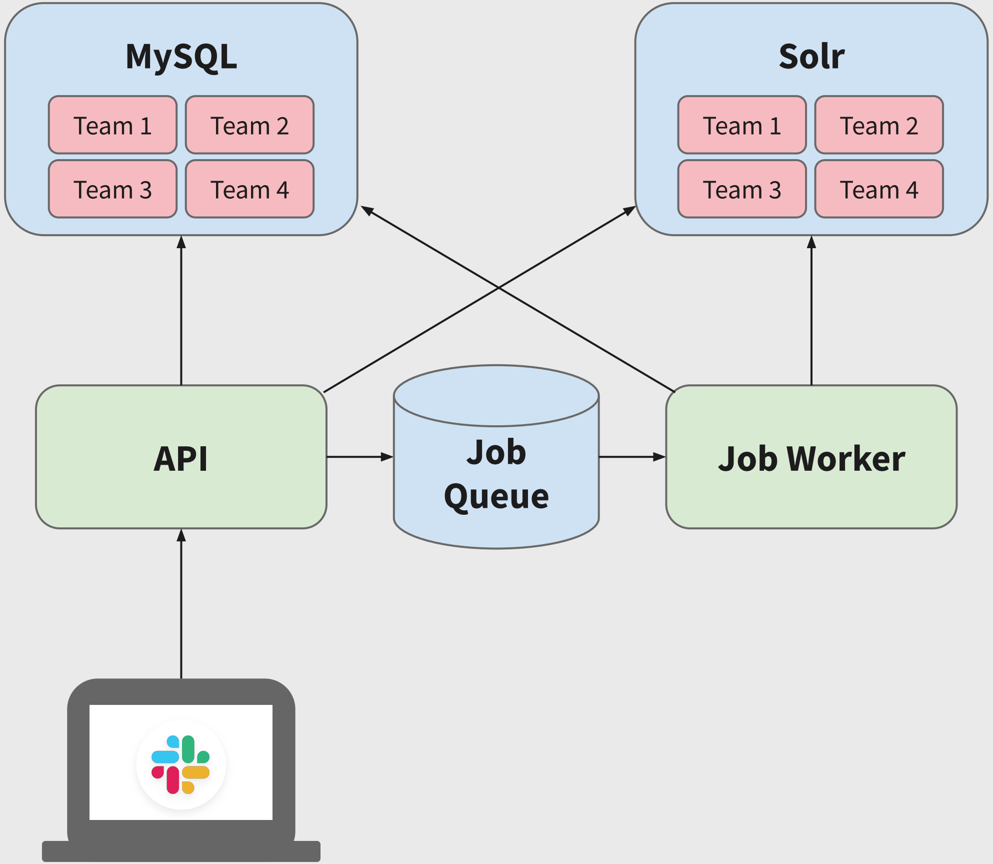
Search Querying

Read Path



Scaling

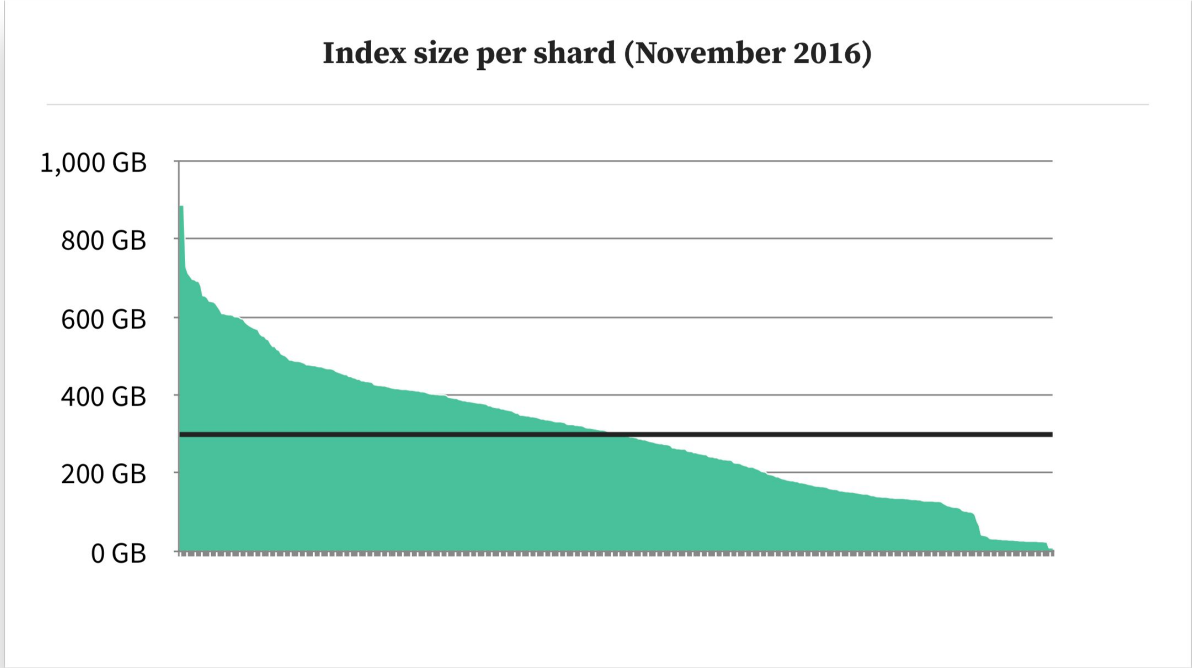
Team Sharding



A Few
Years Later

Indexing

Search performance is getting worse.



Querying

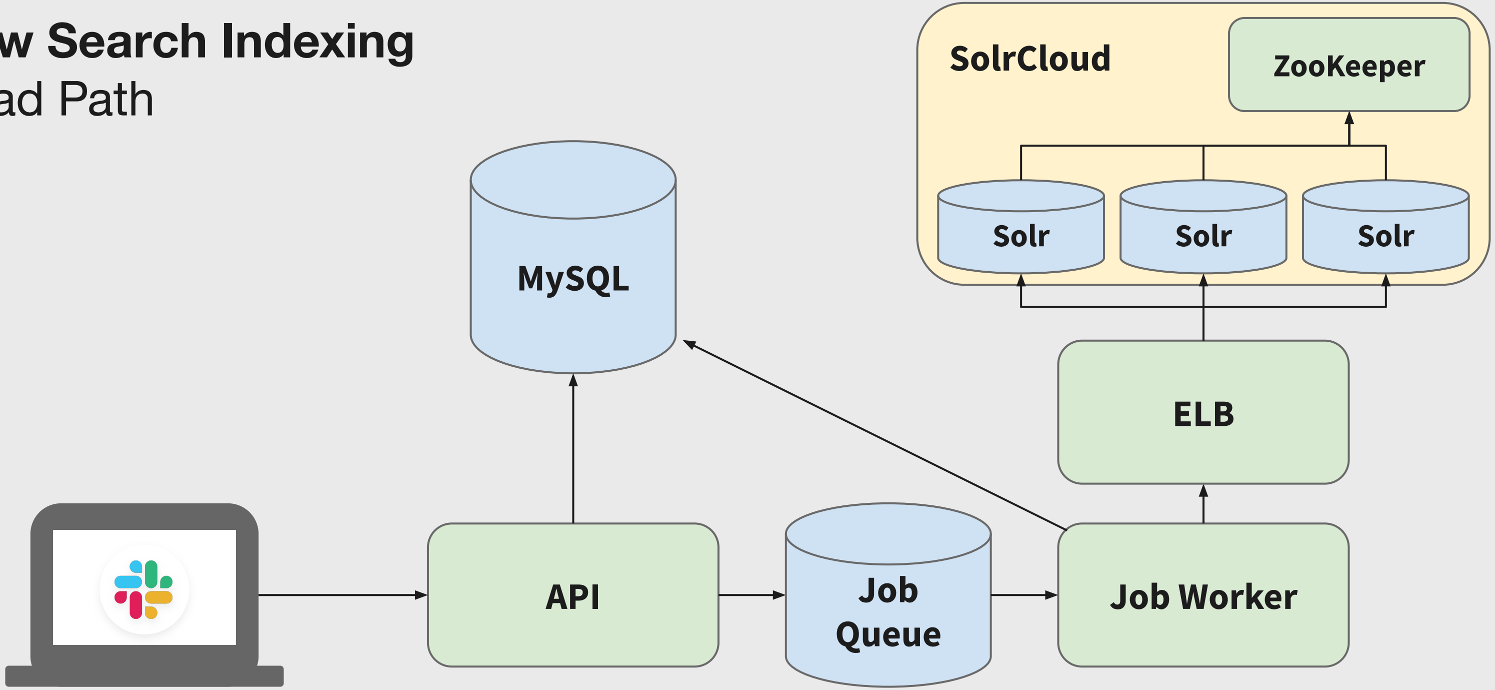
Search quality is getting worse.

Score = TF-IDF

```
log(numDocs / (docFreq + 1)) * sqrt(tf) * (1/sqrt(length))
```

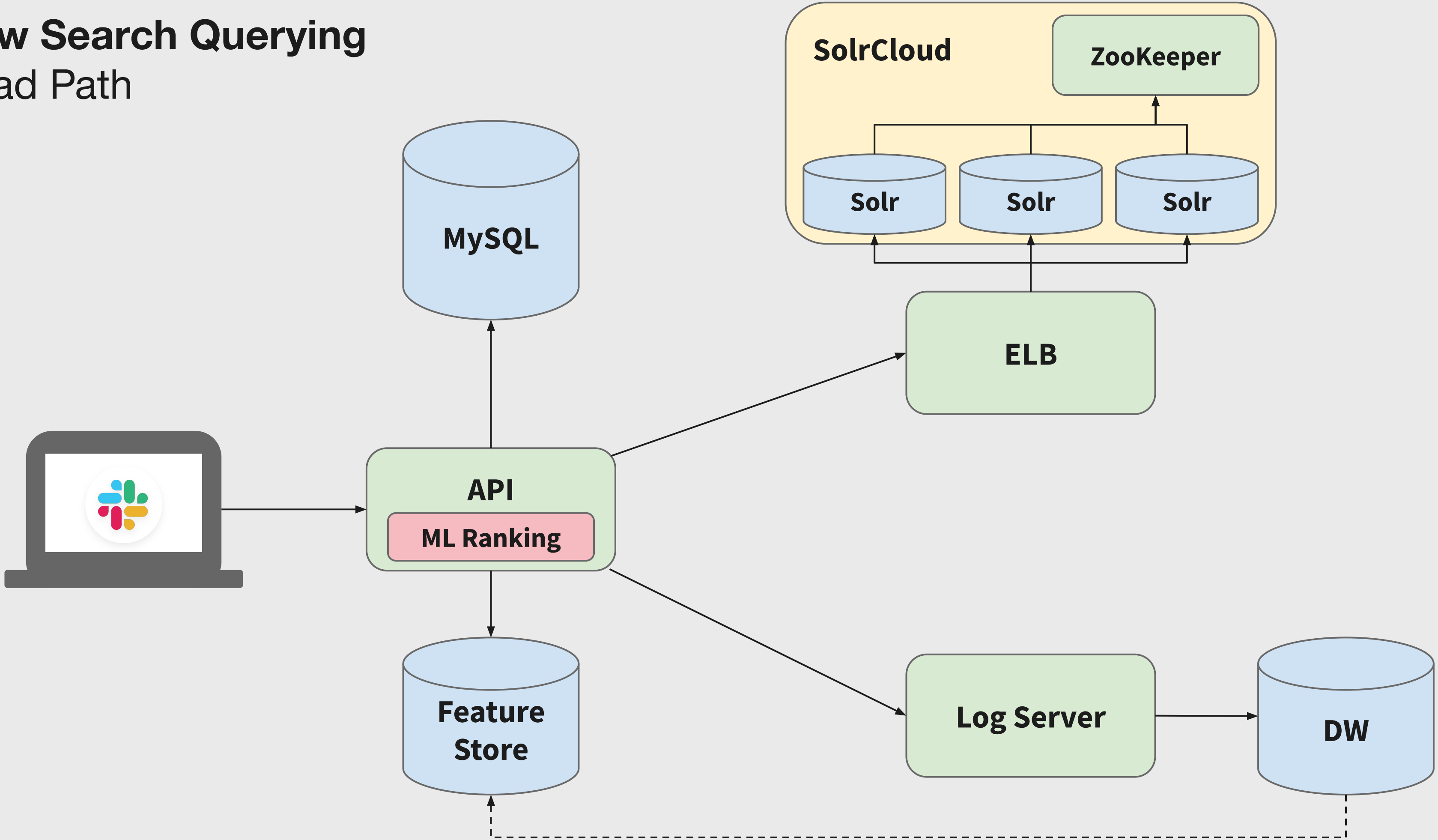
New Search Indexing

Read Path

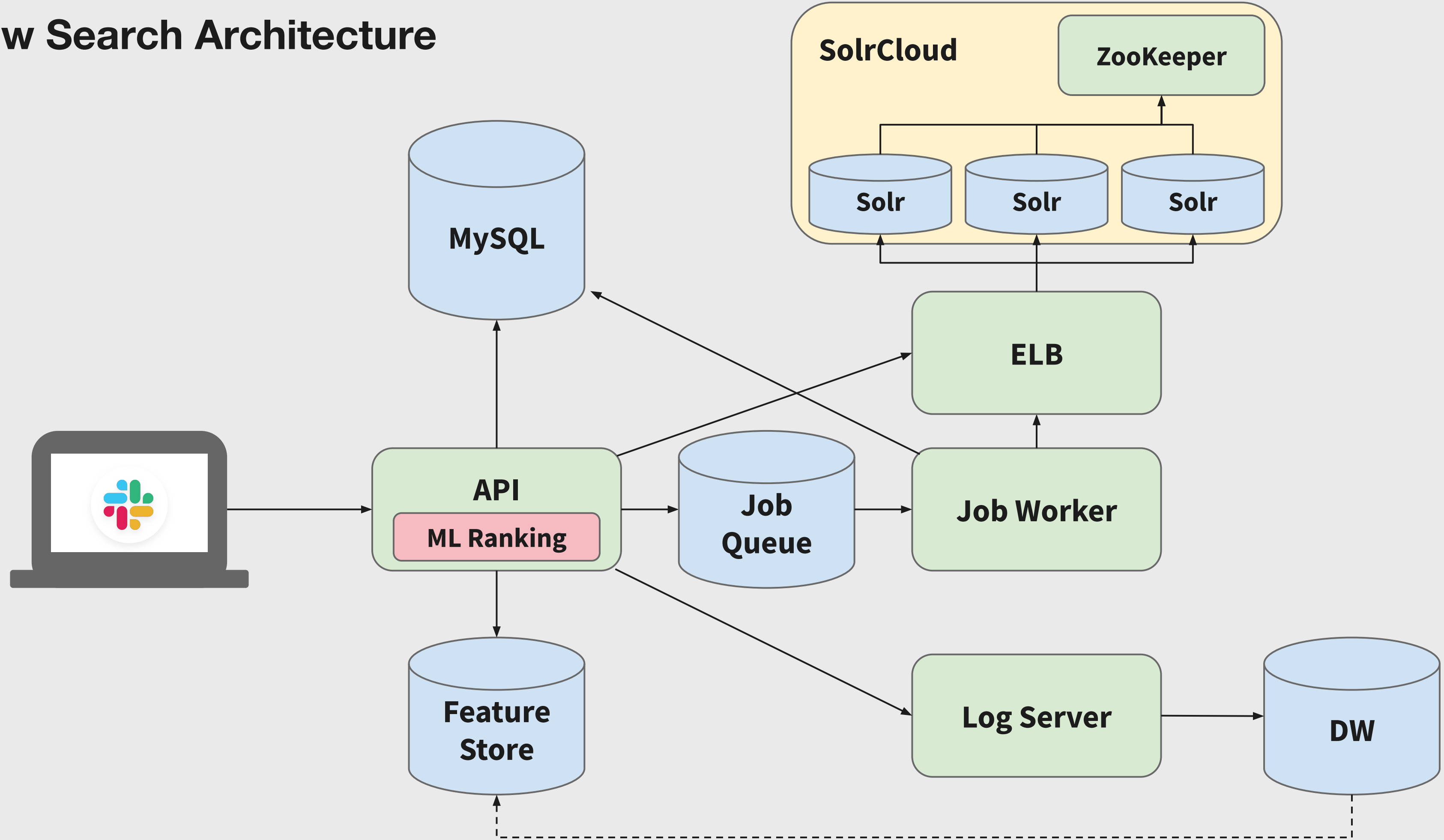


New Search Querying

Read Path

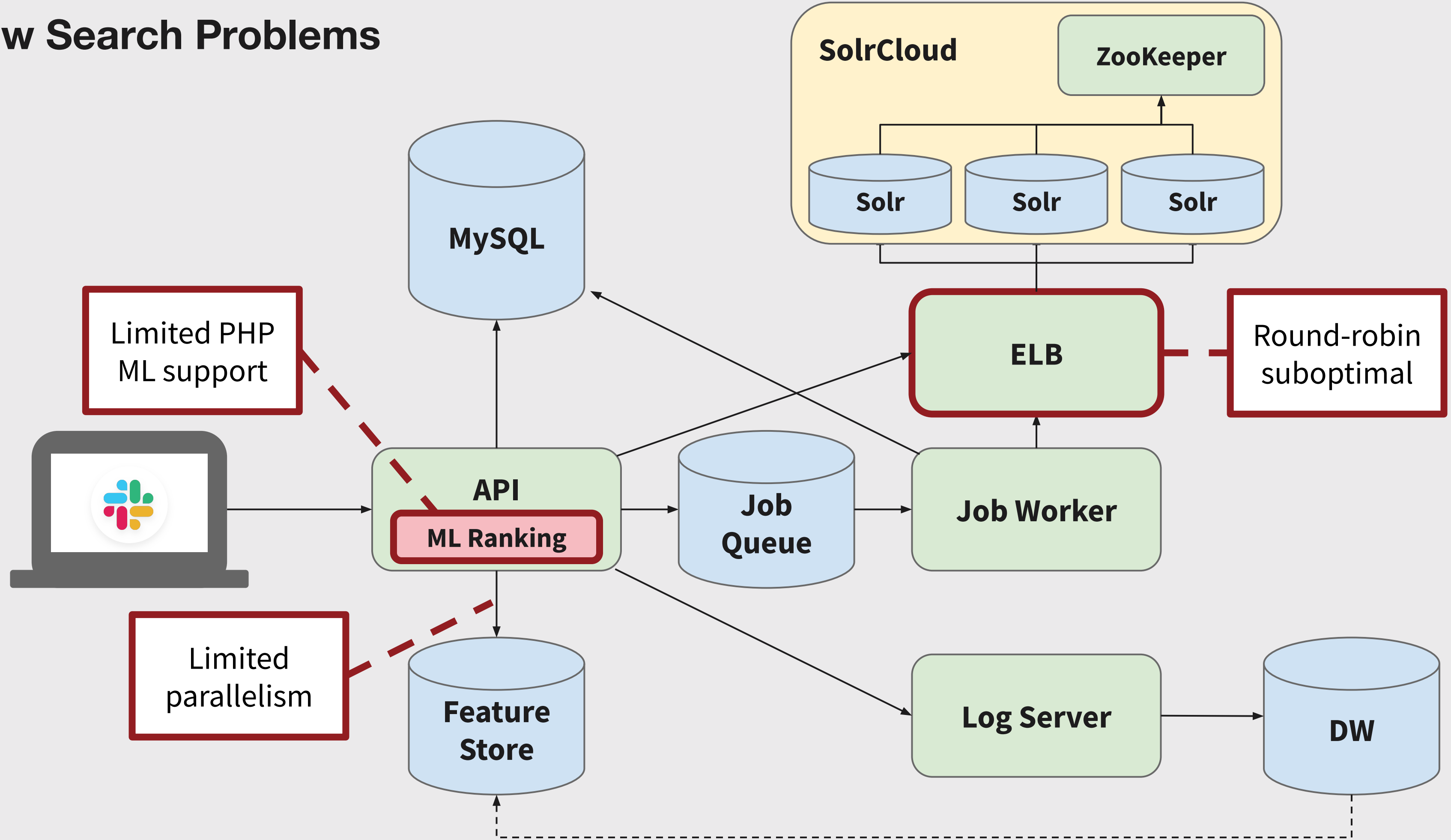


New Search Architecture



A Few
Months Later

New Search Problems

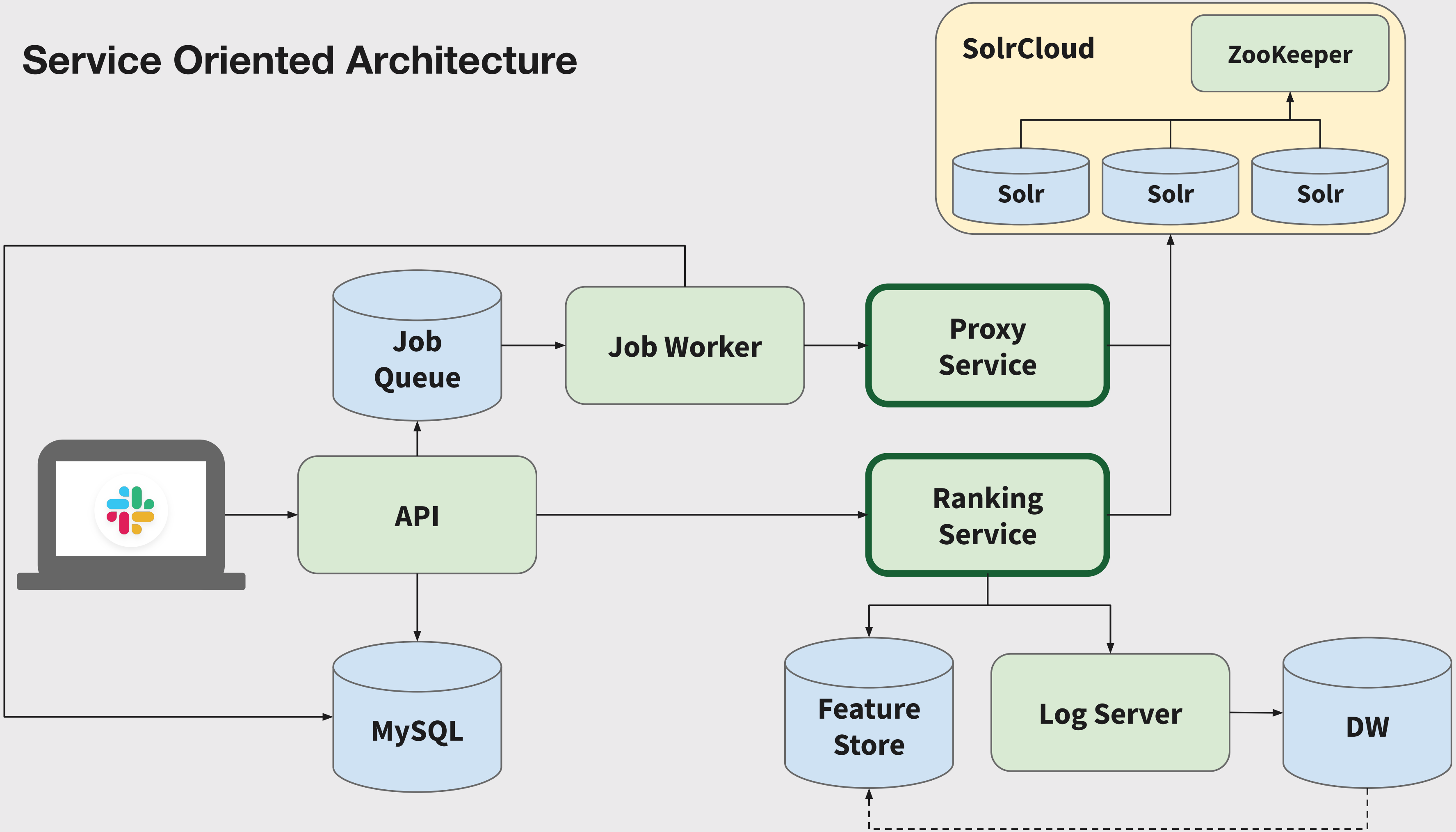


Limited PHP ML support

Round-robin suboptimal

Limited parallelism

Service Oriented Architecture



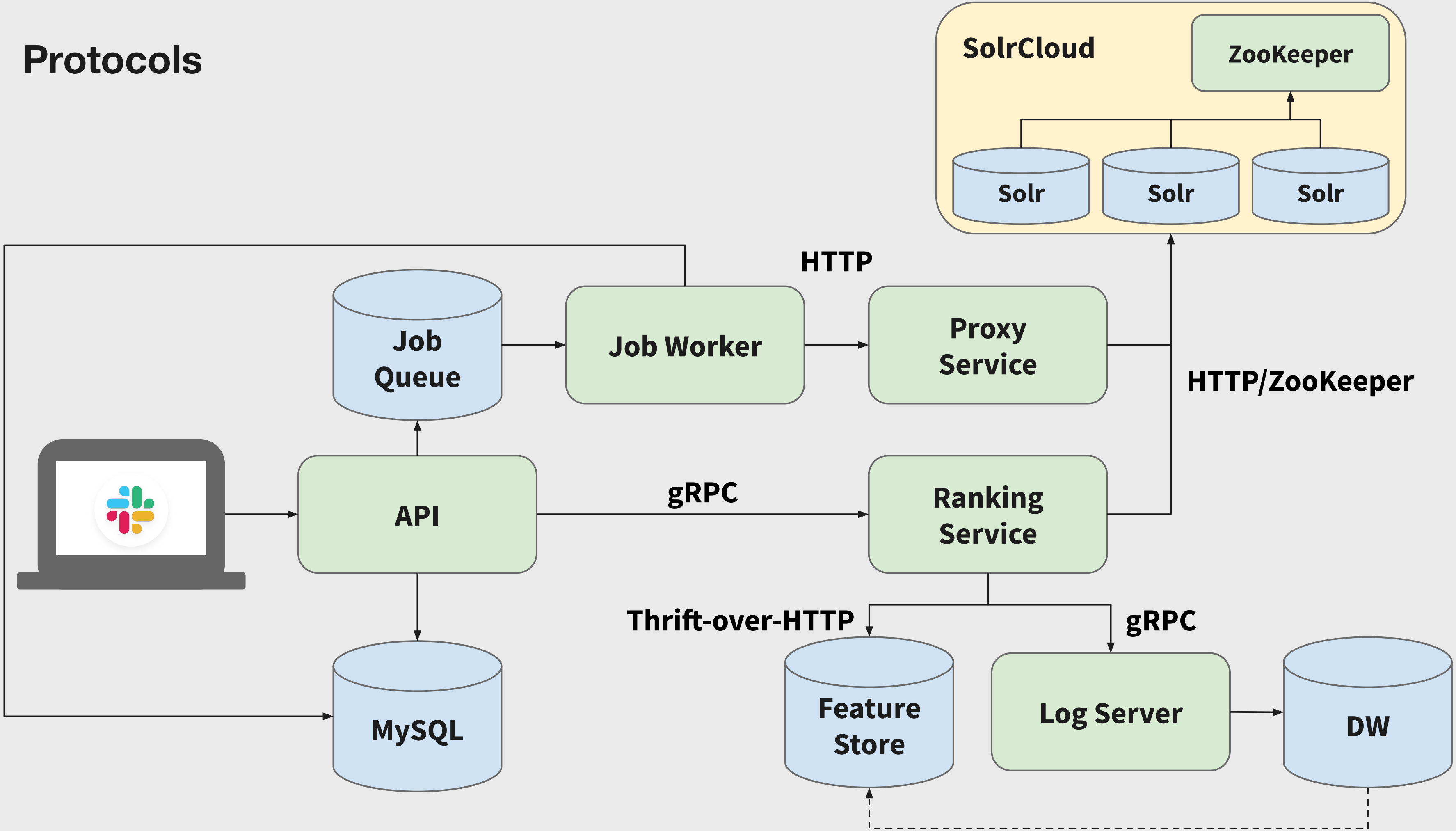
Requirements

- **Written in Java**
  - Solr, SolrJ, XGBoost4J

Requirements

- **Written in Java**
    - Solr, SolrJ, XGBoost4J
- **Supports multiple protocols**
    - HTTP, gRPC, Thrift-over-HTTP

Protocols


SolrCloud
ZooKeeper
Solr
Solr
Solr
HTTP
Job Queue
Job Worker
Proxy Service
HTTP/ZooKeeper
API
gRPC
Ranking Service
Thrift-over-HTTP
gRPC
MySQL
Feature Store
Log Server
DW

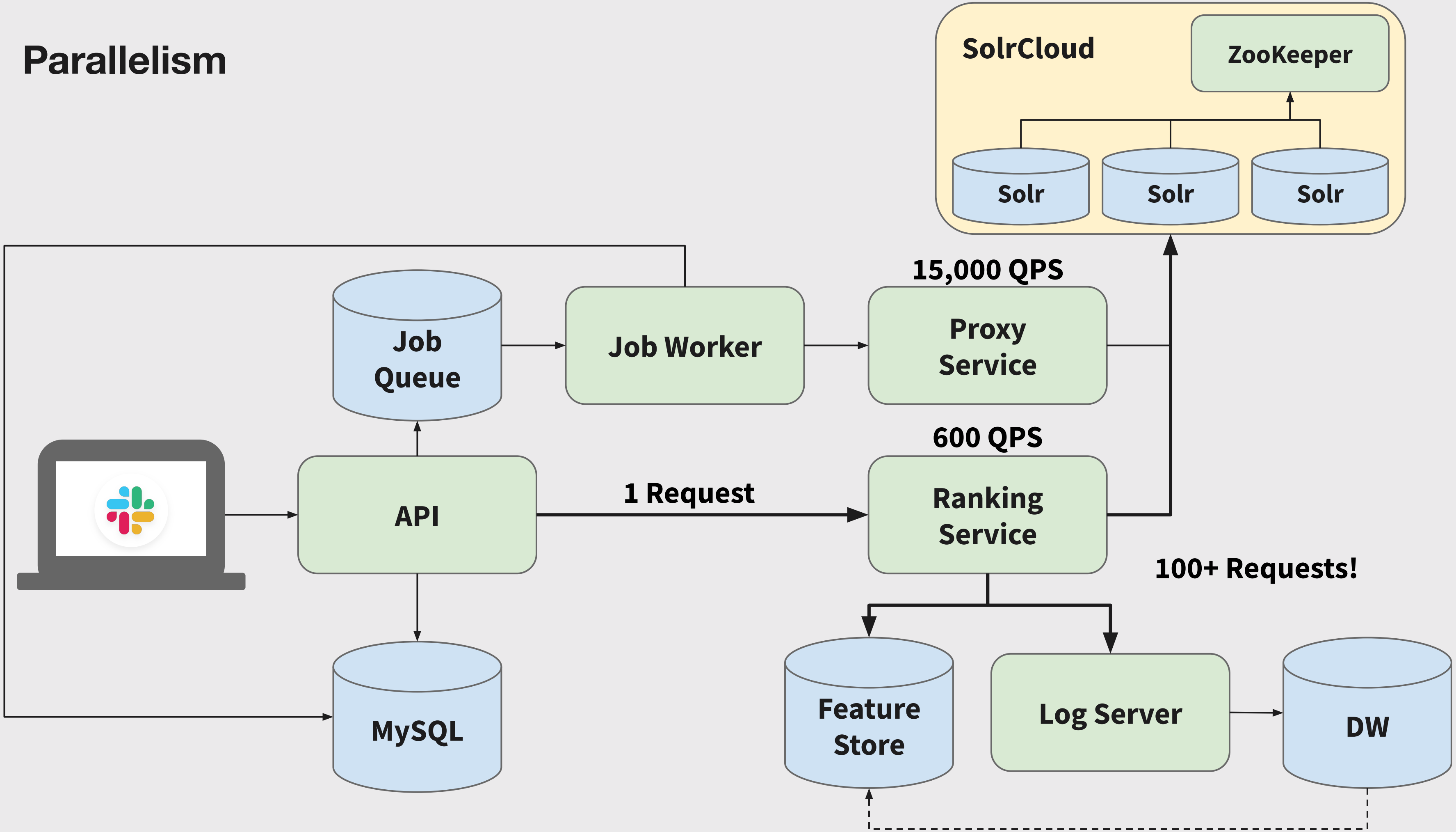
Requirements

- **Written in Java**
  - Solr, SolrJ, XGBoost4J
- **Supports multiple protocols**
  - HTTP, gRPC, Thrift-over-HTTP
- **Integrates with existing tools**
  - Consul, Prometheus, Rsyslog, Honeycomb

Requirements

- **Written in Java**
  - Solr, SolrJ, XGBoost4J
- **Supports multiple protocols**
  - HTTP, gRPC, Thrift-over-HTTP
- **Integrates with existing tools**
  - Consul, Prometheus, Rsyslog, Honeycomb
- **Good parallel performance**

Parallelism



Requirements

- **Written in Java**
  - Solr, SolrJ, XGBoost4J
- **Supports multiple protocols**
  - HTTP, gRPC, Thrift-over-HTTP
- **Integrates with existing tools**
  - Consul, Prometheus, Rsyslog, Honeycomb
- **Good parallel performance**

Armeria

Armeria

Armeria is an open-source asynchronous HTTP/2 RPC/REST client/server library built on top of Java 8, Netty, Thrift and gRPC. Its primary goal is to help engineers build high-performance asynchronous microservices that use HTTP/2 as a session layer protocol.

It is open-sourced and licensed under Apache License 2.0 by LINE Corporation, who uses it in production.

Requirements

- ✅ **Written in Java**
    - Solr, SolrJ, XGBoost4J
- **Supports multiple protocols**
    - HTTP, gRPC, Thrift-over-HTTP
- **Integrates with existing tools**
    - Consul, Prometheus, Rsyslog, Honeycomb
- **Good parallel performance**

Armeria

http://line.github.io/armeria/

Server (HTTP)

```
public class ProxyServlet extends HttpServlet {
  protected void doPost(HttpServletRequest req, HttpServletResponse resp)
      throws ServletException, IOException {
    ...
  }
}
```

```
ServletContextHandler handler = new ServletContextHandler(NO_SESSIONS);
handler.setClassLoader(SolrCloudProxyServer.class.getClassLoader());
handler.addServlet(ProxyServlet.class, "/*");
Server arm = new ServerBuilder()
    .serviceUnder("/solr", new JettyServiceBuilder().handler(handler).build())
    .build();
arm.start();
```

Server (gRPC)

```
service SolrRankService {
  rpc Rank(SolrRankRequest) returns (SolrRankResponse) {}
}
```

```
public class SolrRankService extends SolrRankServiceGrpc.SolrRankServiceImplBase {
  public void rank(SolrRankRequest req, StreamObserver<SolrRankResponse> responseObserver) {
    ...
  }
}
```

```
SolrRankService solrRankService = new SolrRankService(...);
Server arm = new ServerBuilder()
    .service(new GrpcServiceBuilder().addService(solrRankService).build())
    .build();
arm.start();
```

Client (Thrift)

```
service HFileService {
  SingleHFileKeyResponse getValuesSingle(1: SingleHFileKeyRequest req)
  ...
}
```

```
HFileService.Iface client =
    new ClientBuilder("tbinary+http://...")
        .writeTimeout(DEFAULT_CONNECTION_TIMEOUT)
        .decorator(
            RetryingHttpClient.newDecorator(
                RetryStrategy.onServerErrorStatus(DEFAULT_BACKOFF),
                DEFAULT_MAX_TOTAL_ATTEMPTS,
                DEFAULT_RESPONSE_TIMEOUT.toMillis()));
        .build(HFileService.Iface.class);
```

Requirements

- ✅ **Written in Java**
  - Solr, SolrJ, XGBoost4J
- ✅ **Supports multiple protocols**
  - HTTP, gRPC, Thrift-over-HTTP
- **Integrates with existing tools**
  - Consul, Prometheus, Rsyslog, Honeycomb
- **Good parallel performance**

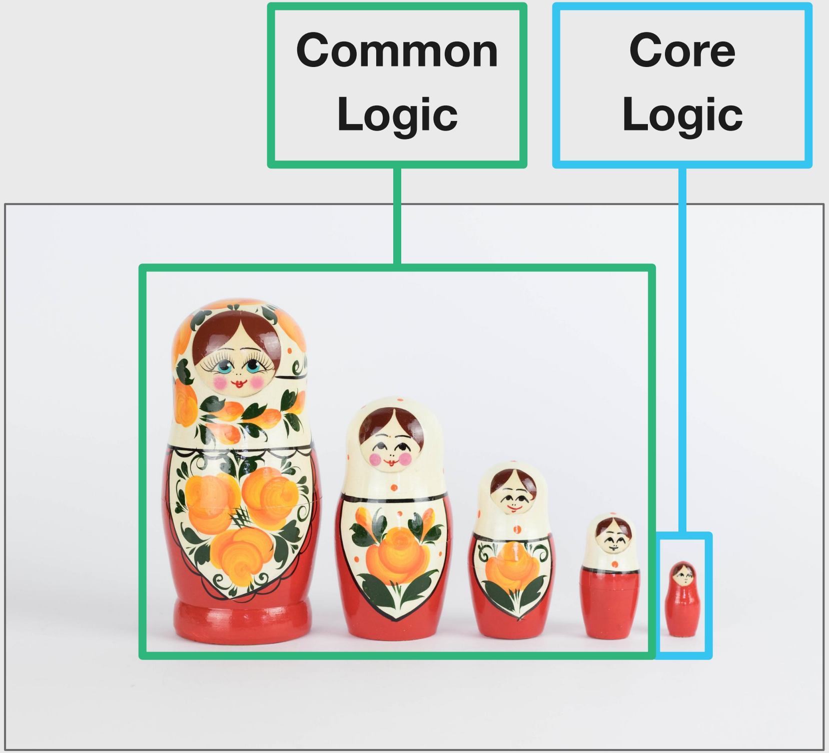
Armeria

http://line.github.io/armeria/

Decorators

```
new ServerBuilder()
    .decorator((delegate, ctx, req) -> {
        // Do something with server request.
        return delegate.execute(ctx, req);
    })
    .build();
```

```
new ClientBuilder(...)
    .decorator((delegate, ctx, req) -> {
        // Do something with client request.
        return delegate.execute(ctx, req);
    })
    .build(...);
```

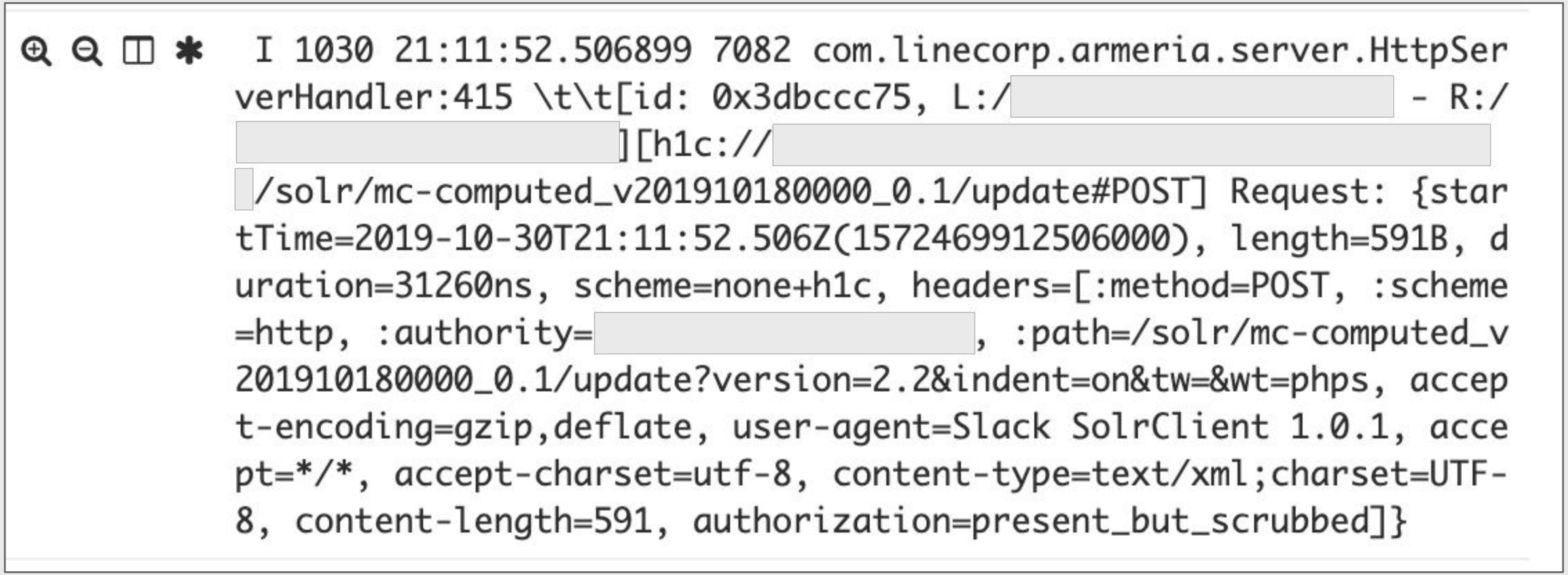


https://www.flickr.com/photos/30478819@N08/41364490961

Logging

```
Server arm = new ServerBuilder()
    ...
    .decorator(
        new LoggingServiceBuilder()
            .requestLogLevel(LogLevel.INFO);
            .successfulResponseLogLevel(LogLevel.INFO)
            .failureResponseLogLevel(LogLevel.ERROR)
            .requestHeadersSanitizer(
                headers -> {
                  if (headers.contains(HttpHeaderNames.AUTHORIZATION)) {
                    return headers
                        .toBuilder()
                        .removeAndThen(HttpHeaderNames.AUTHORIZATION)
                        .add(HttpHeaderNames.AUTHORIZATION, "present_but_scrubbed")
                        .build();
                  }
                  return headers;
                })
            .newDecorator())
    ...
```

Logging

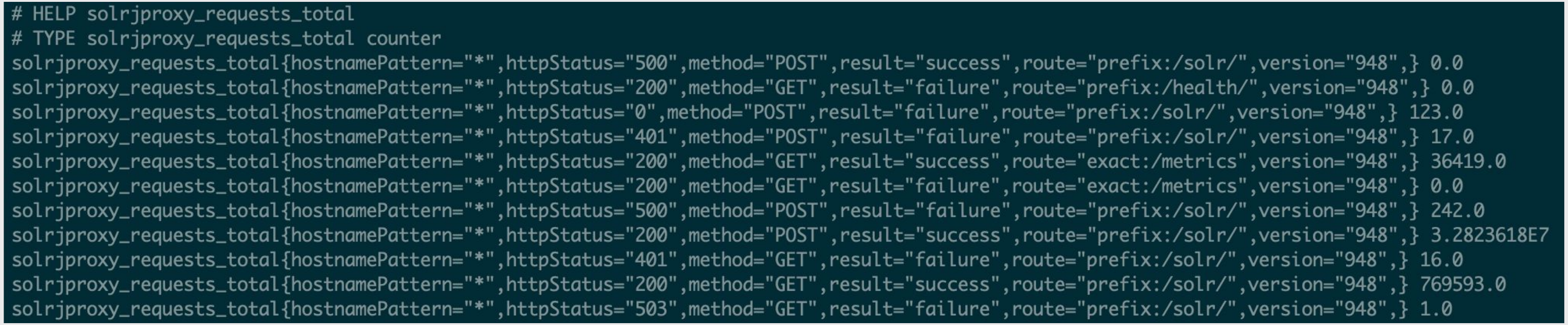


```
 I 1030 21:11:52.506899 7082 com.linecorp.armeria.server.HttpSer
verHandler:415 \t\t[id: 0x3dbccc75, L:/[redacted] - R:/
[redacted]][h1c://[redacted]
[redacted]/solr/mc-computed_v201910180000_0.1/update#POST] Request: {star
tTime=2019-10-30T21:11:52.506Z(1572469912506000), length=591B, d
uration=31260ns, scheme=none+h1c, headers=[:method=POST, :scheme
=http, :authority=[redacted], :path=/solr/mc-computed_v
201910180000_0.1/update?version=2.2&indent=on&tw=&wt=phps, accep
t-encoding=gzip,deflate, user-agent=Slack SolrClient 1.0.1, acce
pt=*/*, accept-charset=utf-8, content-type=text/xml;charset=UTF-
8, content-length=591, authorization=present_but_scrubbed]}
```

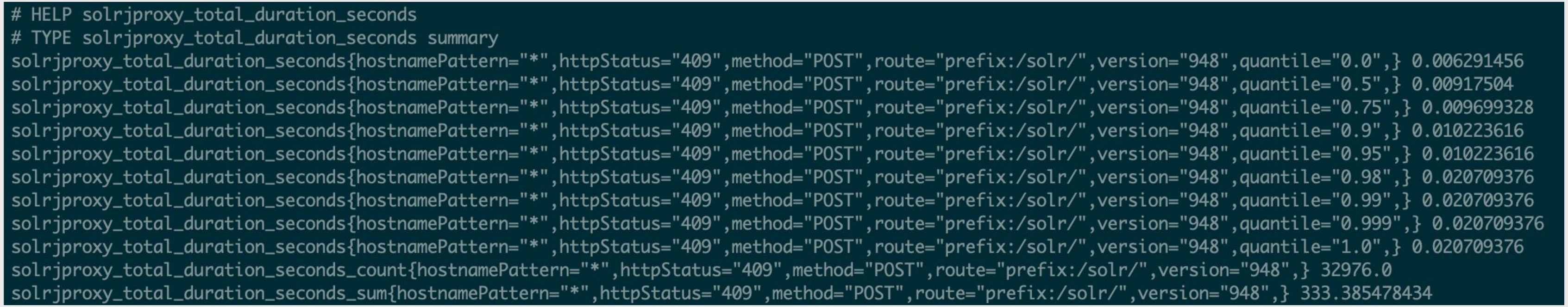
Metrics Prometheus

```
Server arm = new ServerBuilder()
    ...
    .service(
        "/metrics",
        new PrometheusExpositionService(
            config.getMeterRegistry().getPrometheusRegistry())
    )
    ...
    .decorator(
        MetricCollectingService.newDecorator(
            MeterIdPrefixFunction.ofDefault(config.getMeterPrefix())
                .withTags(config.getMeterTags())))
    ...
```

Metrics Prometheus



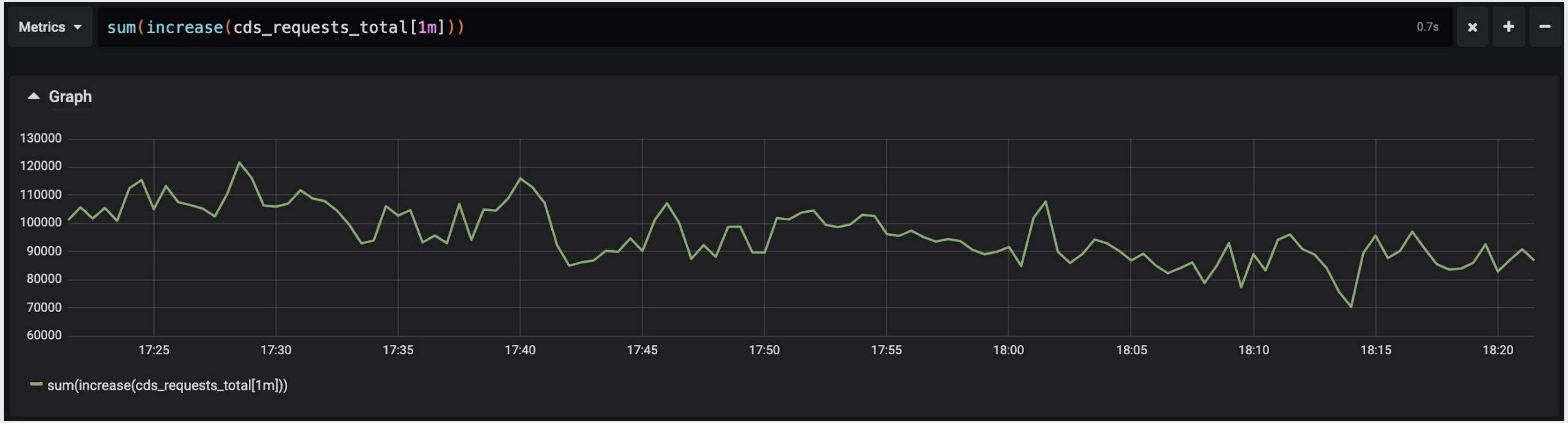
```
# HELP solrjproxy_requests_total
# TYPE solrjproxy_requests_total counter
solrjproxy_requests_total{hostnamePattern="*",httpStatus="500",method="POST",result="success",route="prefix:/solr/",version="948",} 0.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="200",method="GET",result="failure",route="prefix:/health/",version="948",} 0.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="0",method="POST",result="failure",route="prefix:/solr/",version="948",} 123.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="401",method="POST",result="failure",route="prefix:/solr/",version="948",} 17.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="200",method="GET",result="success",route="exact:/metrics",version="948",} 36419.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="200",method="GET",result="failure",route="exact:/metrics",version="948",} 0.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="500",method="POST",result="failure",route="prefix:/solr/",version="948",} 242.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="200",method="POST",result="success",route="prefix:/solr/",version="948",} 3.2823618E7
solrjproxy_requests_total{hostnamePattern="*",httpStatus="401",method="GET",result="failure",route="prefix:/solr/",version="948",} 16.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="200",method="GET",result="success",route="prefix:/solr/",version="948",} 769593.0
solrjproxy_requests_total{hostnamePattern="*",httpStatus="503",method="GET",result="failure",route="prefix:/solr/",version="948",} 1.0
```



```
# HELP solrjproxy_total_duration_seconds
# TYPE solrjproxy_total_duration_seconds summary
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.0",} 0.006291456
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.5",} 0.00917504
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.75",} 0.009699328
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.9",} 0.010223616
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.95",} 0.010223616
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.98",} 0.020709376
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.99",} 0.020709376
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="0.999",} 0.020709376
solrjproxy_total_duration_seconds{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",quantile="1.0",} 0.020709376
solrjproxy_total_duration_seconds_count{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",} 32976.0
solrjproxy_total_duration_seconds_sum{hostnamePattern="*",httpStatus="409",method="POST",route="prefix:/solr/",version="948",} 333.385478434
```

Metrics Prometheus

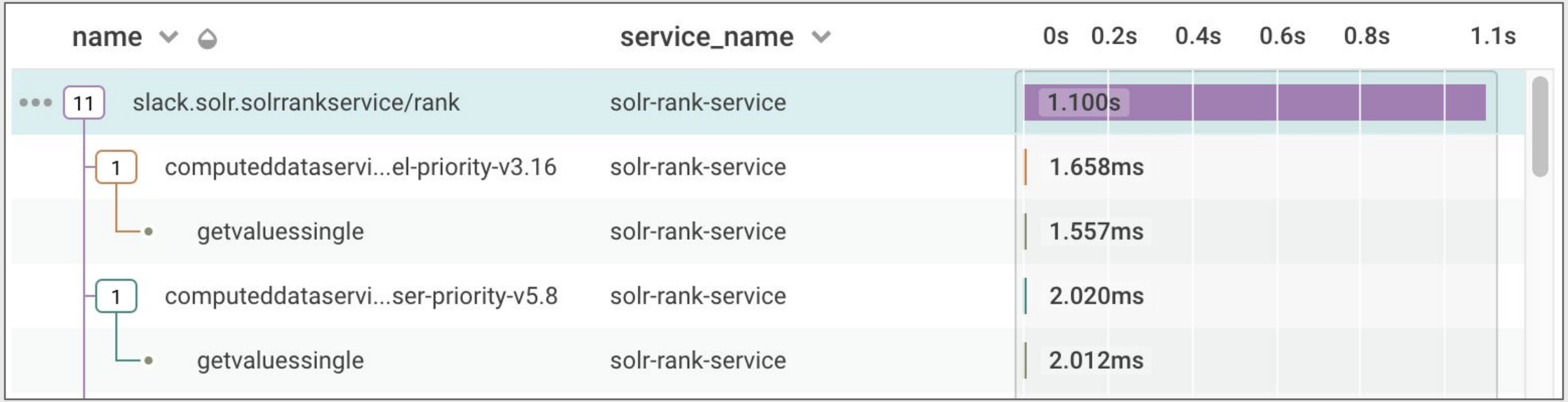
```
HFileService.Iface client =
    new ClientBuilder("tbinary+http://...")
        ...
        .decorator(
            MetricCollectingClient.newDecorator(MeterIdPrefixFunction.ofDefault("cds")))
        ...
```



```
sum(increase(cds_requests_total[1m]))
```

Tracing honeycomb

```
Server arm = new ServerBuilder()
    ...
    .decorator(BraveService.newDecorator(config.getTracing()))
    ...
```



	name	service_name	0s 0.2s 0.4s 0.6s 0.8s 1.1s
11	slack.solr.solrrankservice/rank	solr-rank-service	1.100s
1	computeddataservi...el-priority-v3.16	solr-rank-service	1.658ms
	getvaluessingle	solr-rank-service	1.557ms
1	computeddataservi...ser-priority-v5.8	solr-rank-service	2.020ms
	getvaluessingle	solr-rank-service	2.012ms

Tracing honeycomb

```java
public class MetricCollectingSolrClient extends SolrClient {

  private MetricCollectingSolrClient(SolrClient solrClient, Tracing tracing) {
    this.solrClient = solrClient;
    this.tracing = tracing;
  }

  public NamedList<Object> request(SolrRequest request, String collection)
      throws SolrServerException, IOException {
    Span span = tracing.tracer().newChild(RequestContextCurrentTraceContext.ofDefault().get());
    span.name(getName(collection));
    try (SpanInScope ignore = tracing.tracer().withSpanInScope(span)) {
      try {
        return solrClient.request(request, collection);
      } finally {
        span.finish();
      }
    }
  }
  ...
```

Service Discovery

HashiCorp Consul

```
Server arm = new ServerBuilder()
    ...
    .service("/health", new ManagedHttpHealthCheckService())
    ...
    .serverListener(
       ConsulUpdatingListener.builder(config.getServerName(), config.getPort())
           .serviceId(config.getServerName() + ":" + config.getPort())
           .address(config.getServerAddress())
           .check(String.format("http://localhost:%d/health/", config.getPort()), 10, 5, 15)
           .tag(String.format("version:%s", config.getVersion()))
           .build())
    ...
```

Service Discovery

HashiCorp Consul

```
public class ConsulUpdatingListener extends ServerListenerAdapter {

  public void serverStarted(Server server) throws Exception {
    ImmutableRegistration.Builder registrationBuilder =
        ImmutableRegistration.builder()
            .address(address)
            .port(port)
            .checks(checks)
            .name(service)
            .id(serviceId)
            .addTags(tags)
    }
    agentClient.register(registrationBuilder.build());
  }

  public void serverStopping(Server server) throws Exception {
    agentClient.deregister(serviceId);
    Thread.sleep(15000);
  }
  ...
```

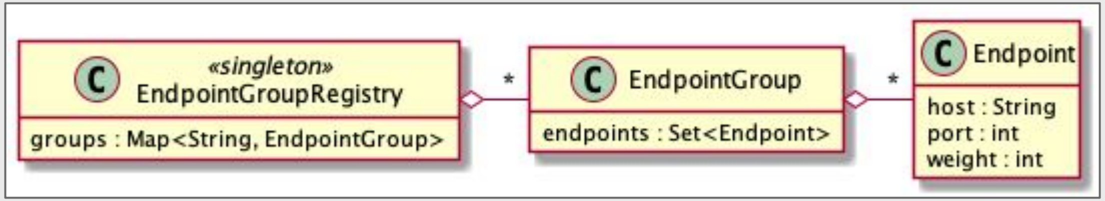
Service Discovery

Why sleep on stop? **To drain requests!**

```
Runtime.getRuntime()
    .addShutdownHook(
        new Thread() {
          @Override
          public void start() {
            log.info("Shutdown hook called");
            arm.stop();
          }
        });
```

Service Discovery

HashiCorp Consul



«singleton»
EndpointGroupRegistry
groups : Map<String, EndpointGroup>

*

EndpointGroup
endpoints : Set<Endpoint>

*

Endpoint
host : String
port : int
weight : int

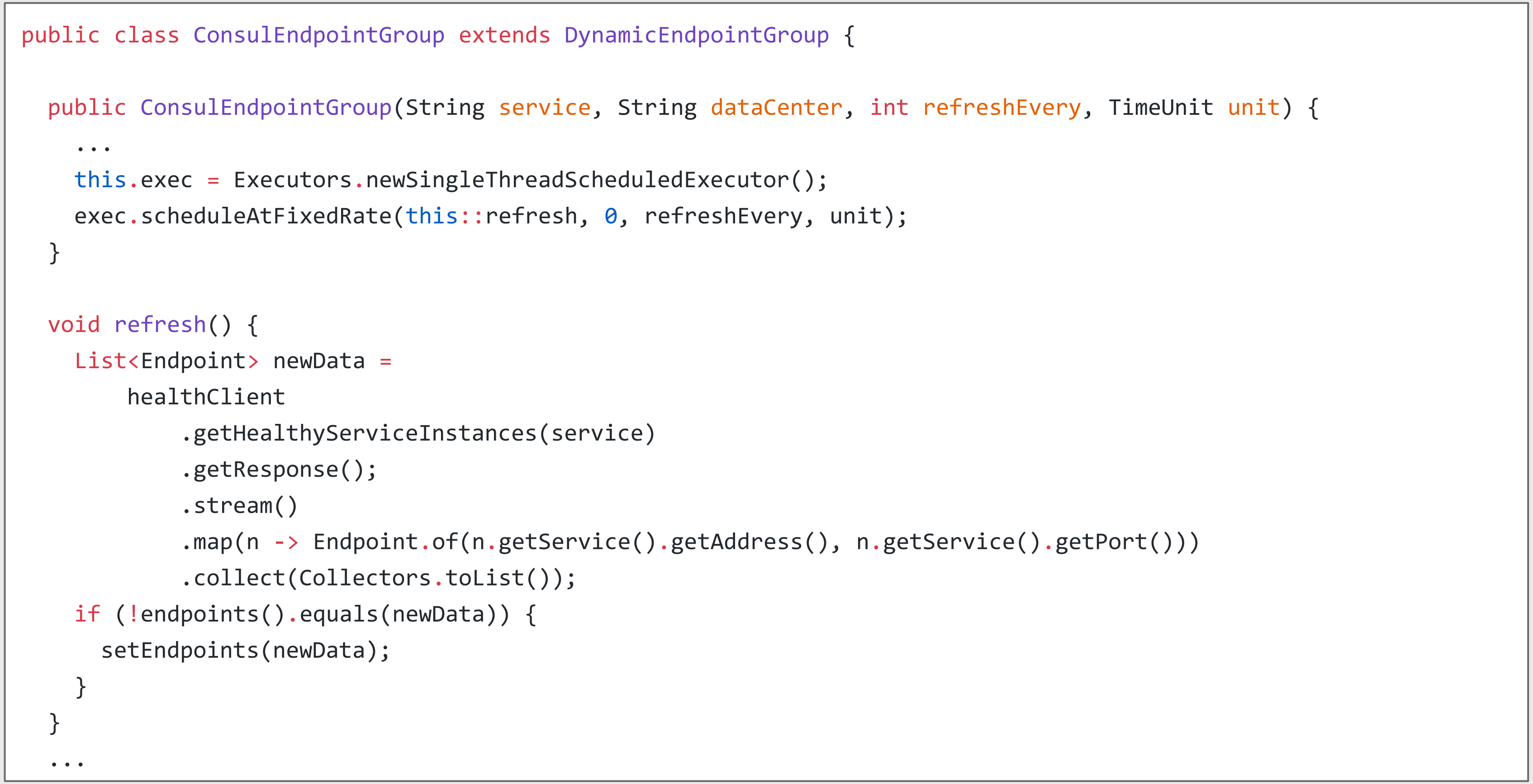
https://line.github.io/armeria/client-service-discovery.html

```
public static void registerEndpoint(
    String service, String partition, int refreshEvery, TimeUnit unit) {
  EndpointGroupRegistry.register(
      service,
      new ConsulEndpointGroup(service, partition, refreshEvery, unit),
      EndpointSelectionStrategy.ROUND_ROBIN);
}
```

```
new ClientBuilder(String.format("tbinary+http://group:%s/", service))
```

Service Discovery

HashiCorp Consul



```
public class ConsulEndpointGroup extends DynamicEndpointGroup {

  public ConsulEndpointGroup(String service, String dataCenter, int refreshEvery, TimeUnit unit) {
    ...
    this.exec = Executors.newSingleThreadScheduledExecutor();
    exec.scheduleAtFixedRate(this::refresh, 0, refreshEvery, unit);
  }

  void refresh() {
    List<Endpoint> newData =
        healthClient
            .getHealthyServiceInstances(service)
            .getResponse();
            .stream()
            .map(n -> Endpoint.of(n.getService().getAddress(), n.getService().getPort()))
            .collect(Collectors.toList());
    if (!endpoints().equals(newData)) {
      setEndpoints(newData);
    }
  }
  ...
```

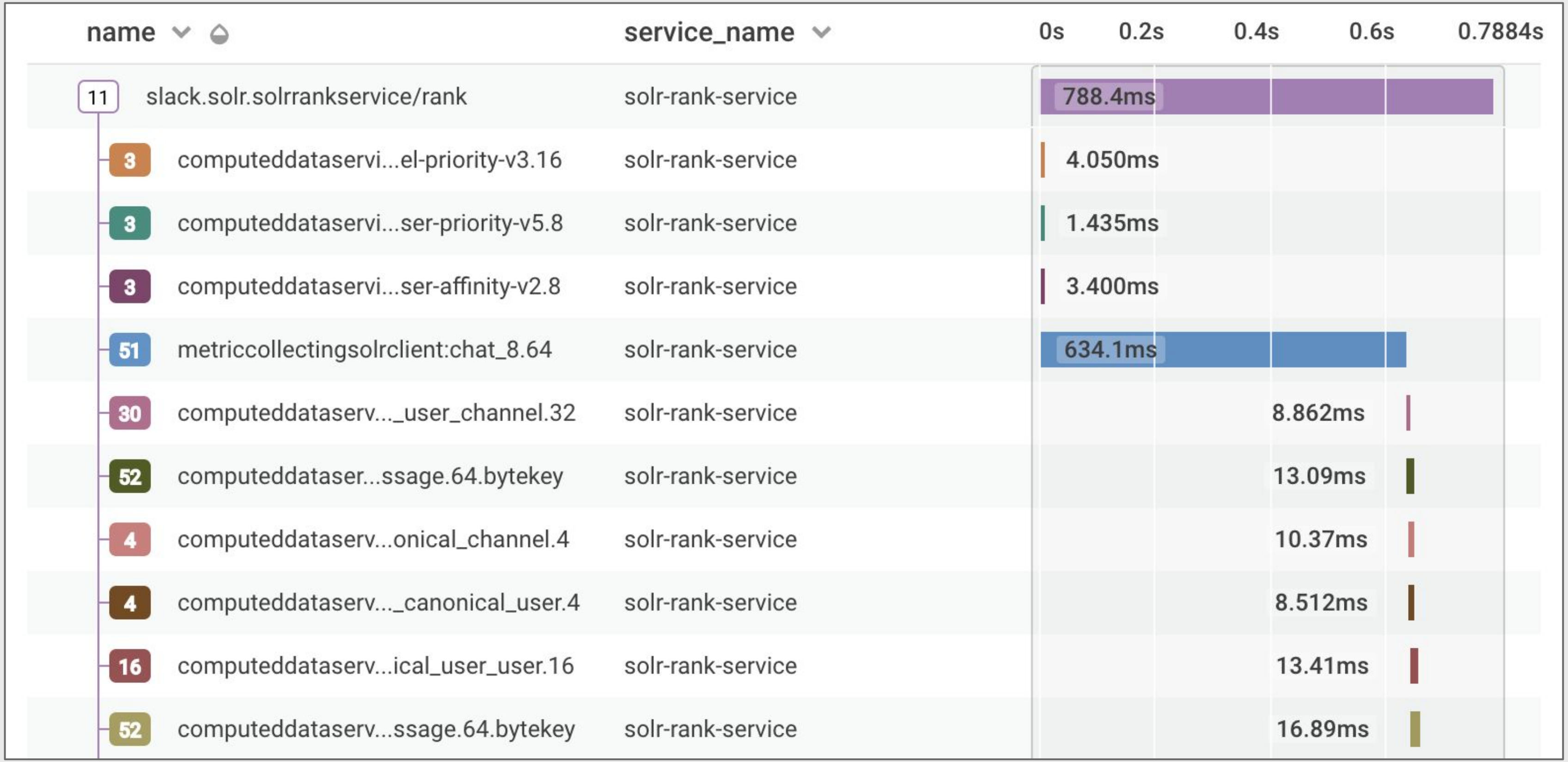
Requirements

- ✅ **Written in Java**
  - ○ Solr, SolrJ, XGBoost4J
- ✅ **Supports multiple protocols**
  - ○ HTTP, gRPC, Thrift-over-HTTP
- ✅ **Integrates with existing tools**
  - ○ Consul, Prometheus, Rsyslog, Honeycomb
- ● **Good parallel performance**

Armeria

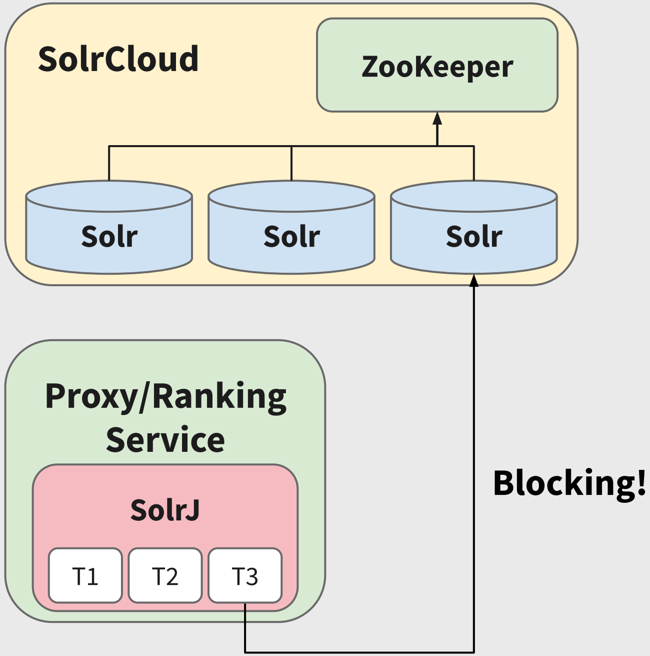
http://line.github.io/armeria/

Performance



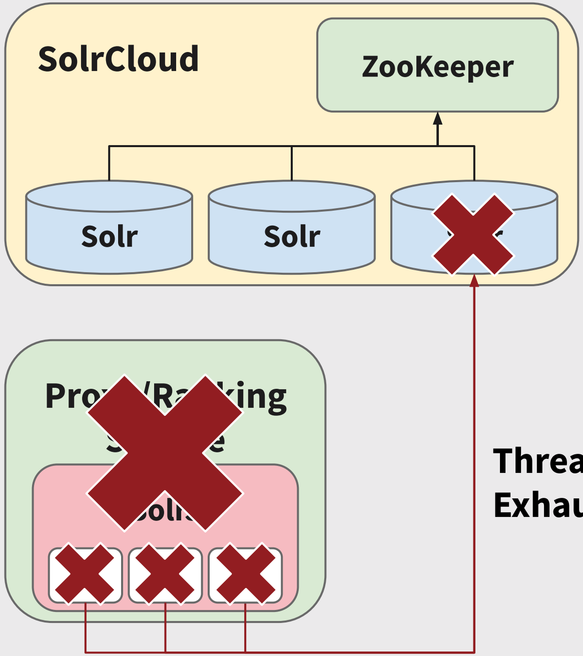
	name	service_name	0s 0.2s 0.4s 0.6s 0.7884s
11	slack.solr.solrrankservice/rank	solr-rank-service	788.4ms
3	computeddataservi...el-priority-v3.16	solr-rank-service	4.050ms
3	computeddataservi...ser-priority-v5.8	solr-rank-service	1.435ms
3	computeddataservi...ser-affinity-v2.8	solr-rank-service	3.400ms
51	metriccollectingsolrclient:chat_8.64	solr-rank-service	634.1ms
30	computeddataserv..._user_channel.32	solr-rank-service	8.862ms
52	computeddataser...ssage.64.bytekey	solr-rank-service	13.09ms
4	computeddataserv...onical_channel.4	solr-rank-service	10.37ms
4	computeddataserv..._canonical_user.4	solr-rank-service	8.512ms
16	computeddataserv...ical_user_user.16	solr-rank-service	13.41ms
52	computeddataserv...ssage.64.bytekey	solr-rank-service	16.89ms

Blocking SolrJ



```
ServiceRequestContext ctx = RequestContext.current();
ctx.blockingTaskExecutor().submit(() -> {
  ...
});
```

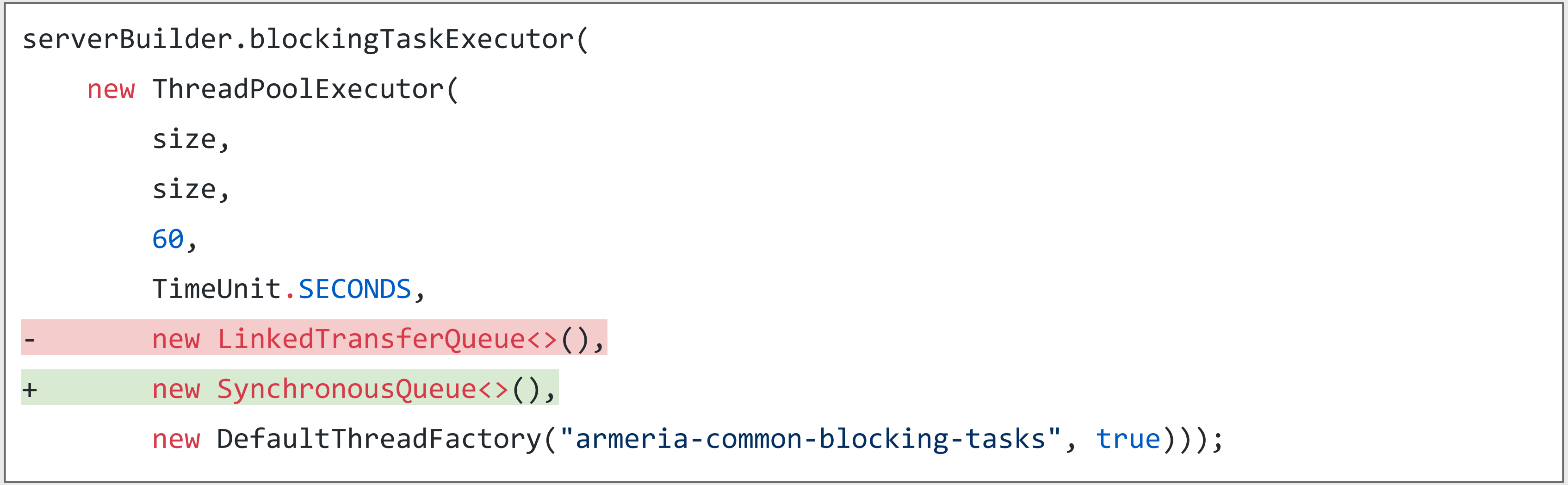
Blocking SolrJ



ThreadPool Exhaustion!

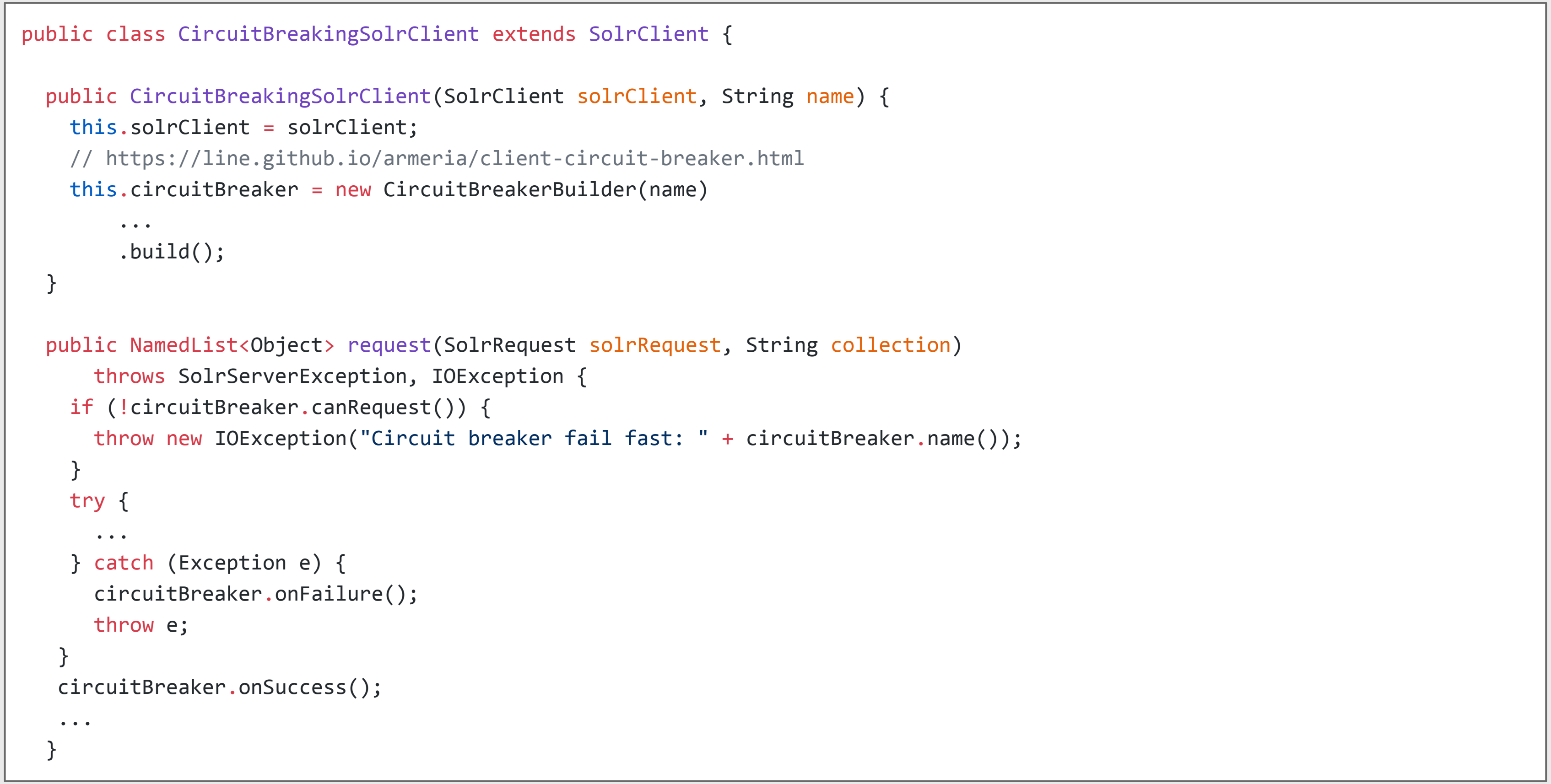
```
ServiceRequestContext ctx = RequestContext.current();
ctx.blockingTaskExecutor([illegible]t(() -> {
  ...
});
```

Blocking SolrJ



```
serverBuilder.blockingTaskExecutor(
  new ThreadPoolExecutor(
    size,
    size,
    60,
    TimeUnit.SECONDS,
-   new LinkedTransferQueue<>(),
+   new SynchronousQueue<>(),
    new DefaultThreadFactory("armeria-common-blocking-tasks", true)));
```

Blocking SolrJ



```
public class CircuitBreakingSolrClient extends SolrClient {

  public CircuitBreakingSolrClient(SolrClient solrClient, String name) {
    this.solrClient = solrClient;
    // https://line.github.io/armeria/client-circuit-breaker.html
    this.circuitBreaker = new CircuitBreakerBuilder(name)
        ...
        .build();
  }

  public NamedList<Object> request(SolrRequest solrRequest, String collection)
      throws SolrServerException, IOException {
    if (!circuitBreaker.canRequest()) {
      throw new IOException("Circuit breaker fail fast: " + circuitBreaker.name());
    }
    try {
      ...
    } catch (Exception e) {
      circuitBreaker.onFailure();
      throw e;
   }
   circuitBreaker.onSuccess();
   ...
  }
```

Requirements

✅ **Written in Java**

- Solr, SolrJ, XGBoost4J

✅ **Supports multiple protocols**

- HTTP, gRPC, Thrift-over-HTTP

✅ **Integrates with existing tools**

- Consul, Prometheus, Rsyslog, Honeycomb

✅ **Good parallel performance**

Armeria

http://line.github.io/armeria/

What’s Next?

Next steps

- More visibility (logging, metrics, tracing)
- Async Solr client
- Integration with Envoy Service Mesh
- Deployment on Kubernetes

Thank you!
Questions?