

Running a Serverless Lucene Reverse Geocoder

Alexander Reelsen
alex@elastic.co
@spinscale

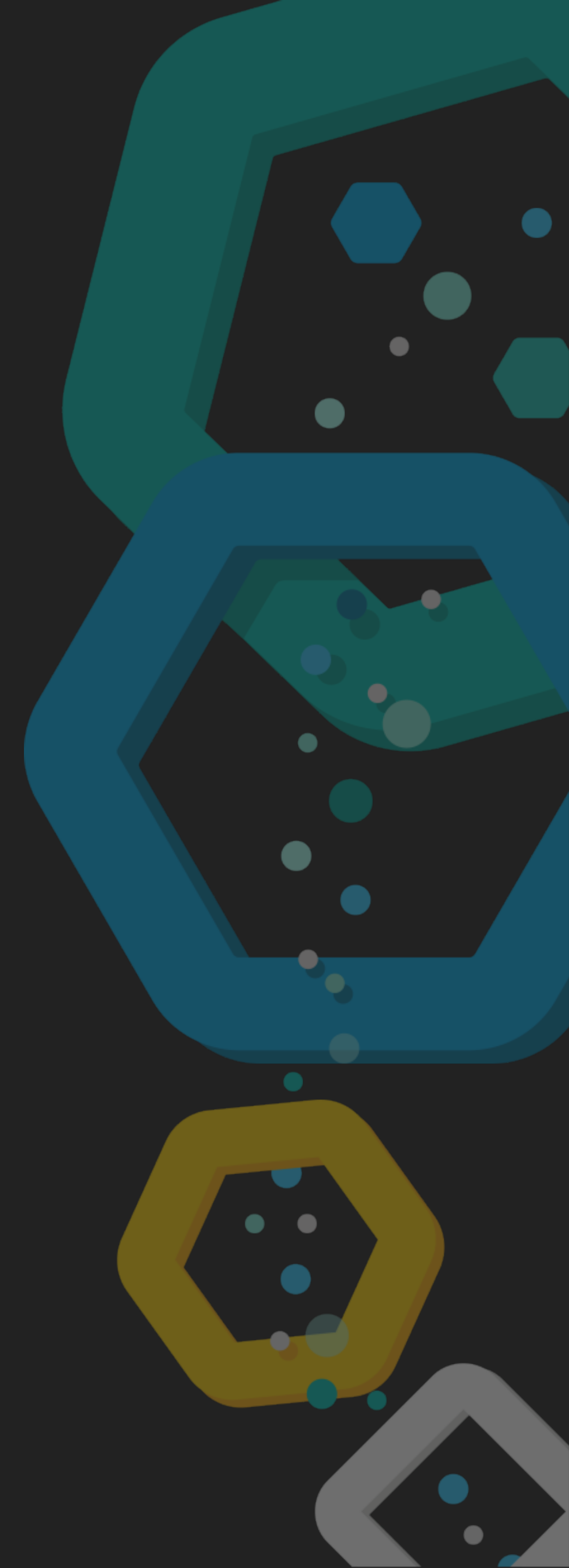


Agenda

- ▶ What is serverless?
- ▶ Searching for Locations
- ▶ Demo
- ▶ How to execute java code faster



Serverless?



Serverless?

- ▶ FaaS (Function as a Service)
- ▶ Execution Environment as a Service
- ▶ Payment model: Pay per code runtime
- ▶ Not running? No bill!
- ▶ Configure memory size (also changes CPU power)
- ▶ Maximum function execution time
- ▶ Provider takes care of scaling functions



Examples

- ▶ Good: Short lived HTTP requests
- ▶ Good: Short running jobs
- ▶ Good: Event streaming & processing
- ▶ Good: Share nothing web applications
- ▶ Good: Parallelizable workloads
- ▶ Bad: Slack bots?



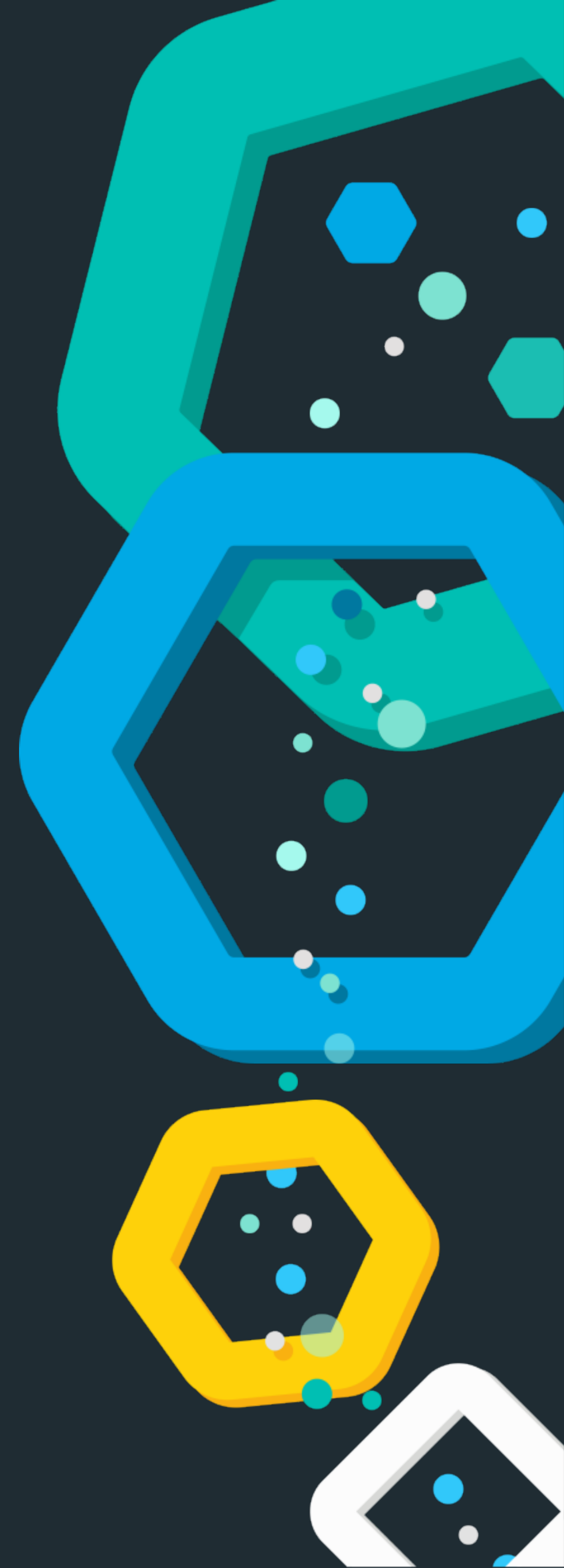
Providers?

- ▶ AWS Lambda, GCP Cloud Functions, Azure Cloud Functions, Cloudflare, IBM OpenWhisk, Google Cloud Run
- ▶ FaaS: FaaS, Binaris, Spotinst
- ▶ K8s: KNative, Fission, Kubeless, Nuclio, OpenFaaS
- ▶ Docker: Fn, OpenFaaS

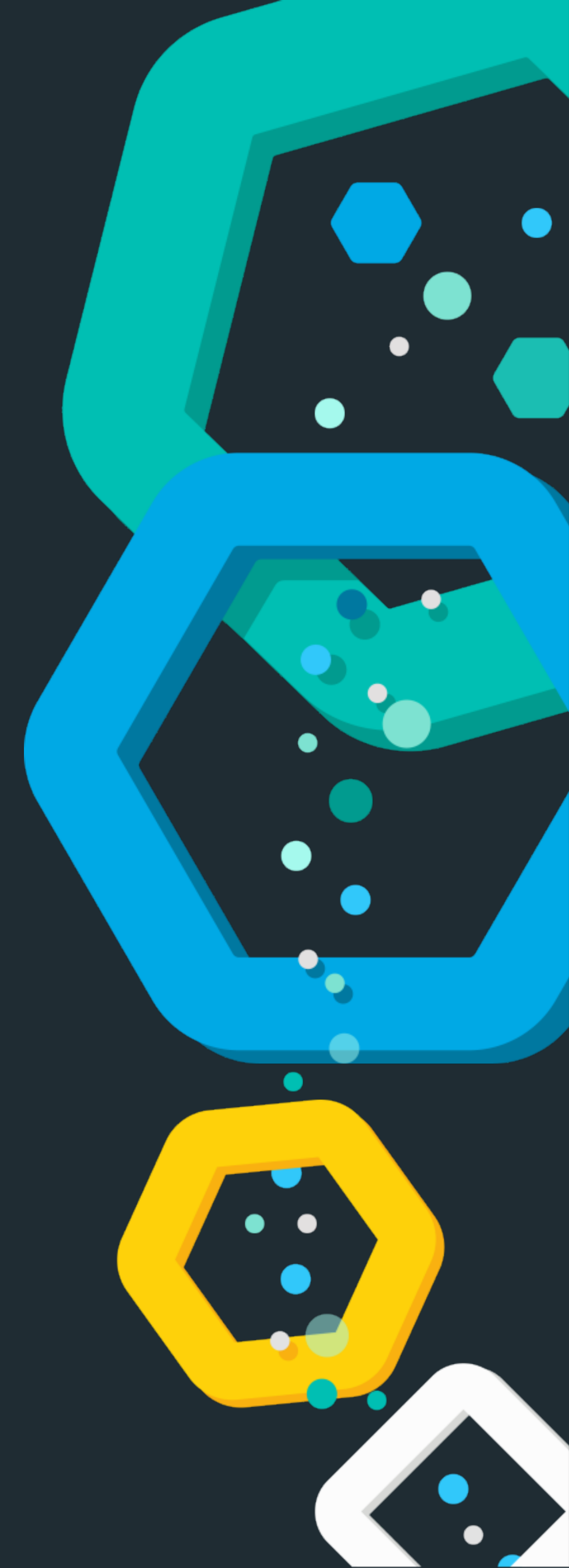


Java?

- ▶ Not too well suited for short lived tasks
- ▶ JVM startup time
- ▶ JIT compiler
- ▶ Dependency initialisation
- ▶ Application initialisation

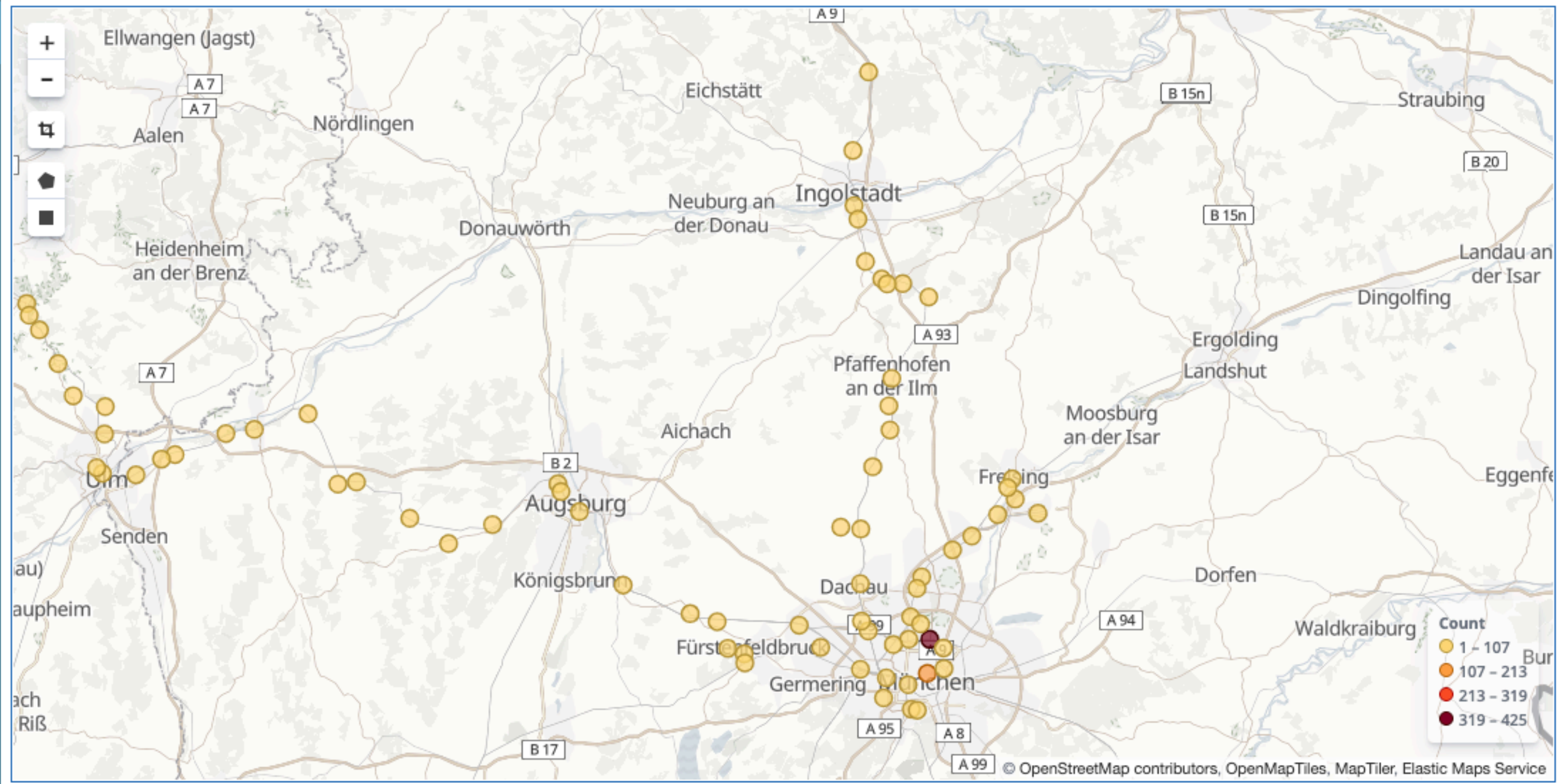


Location Tracker: Owntracks + Lambda + Kibana



Add a filter +

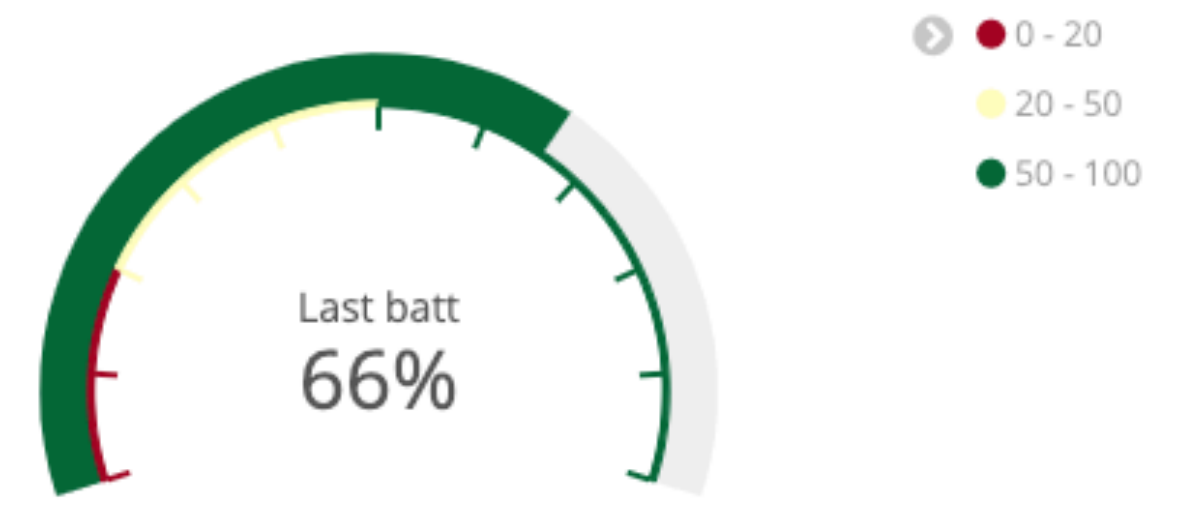
Last Positions



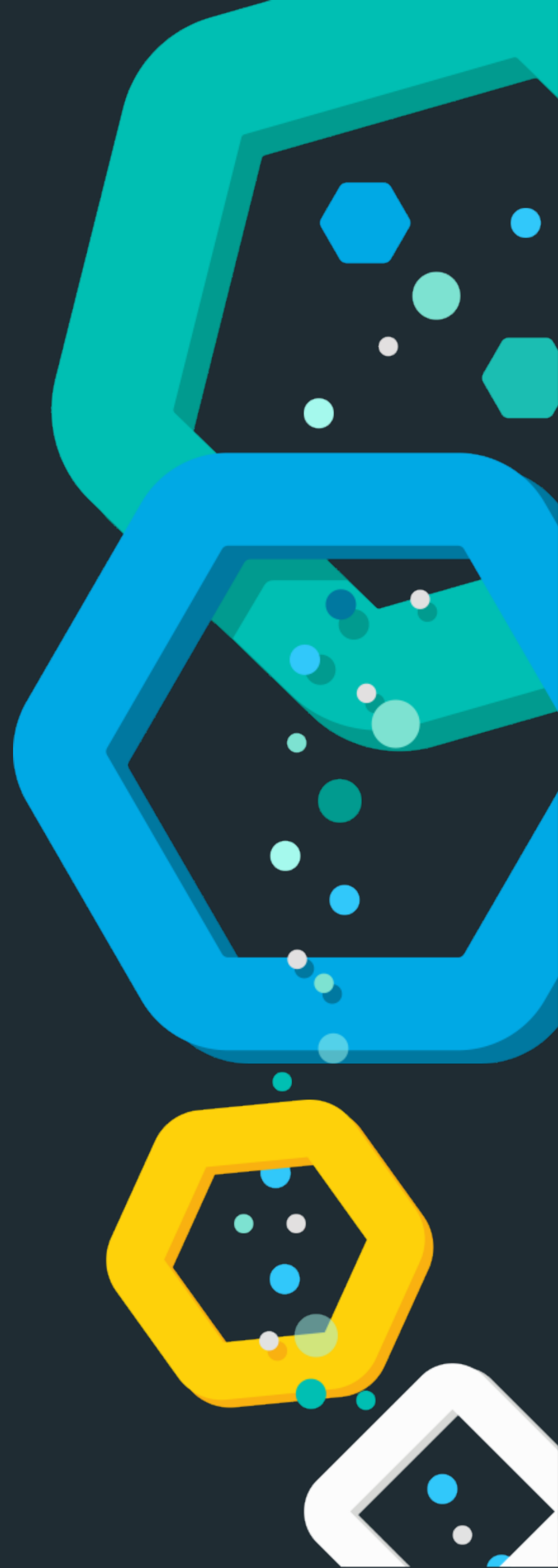
Last update

November 18th 2019, 09:19:47.000
Last update

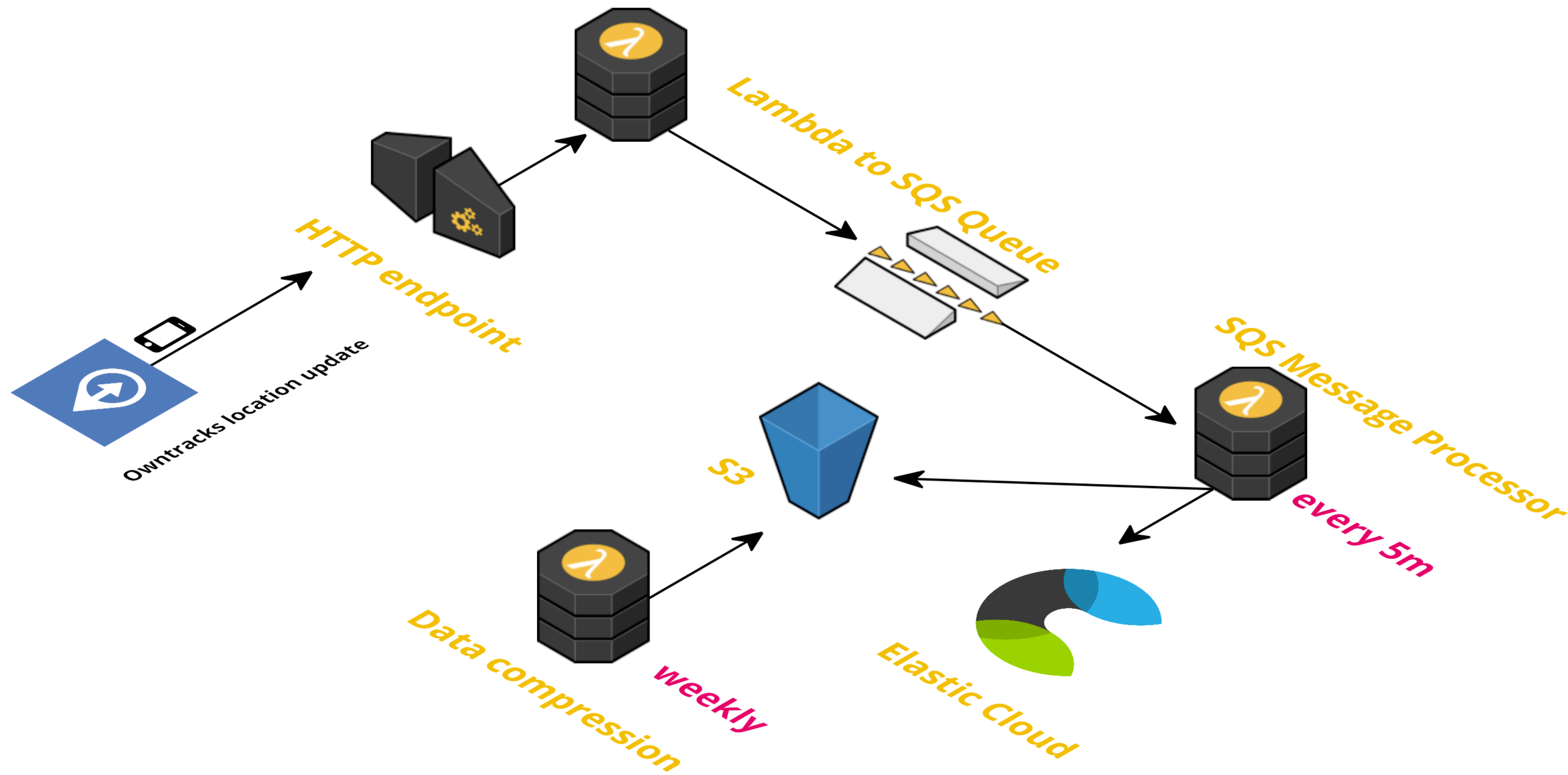
Last battery indicator



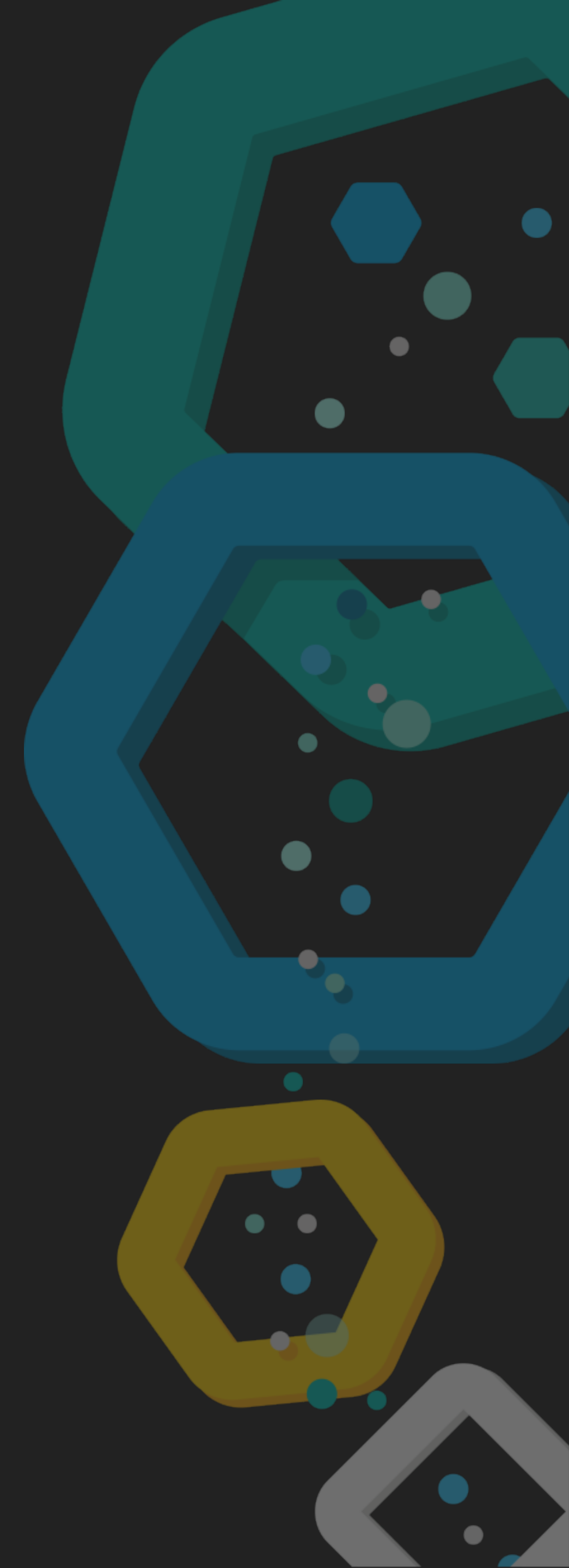
Battery over time



<https://github.com/spinscale/serverless-owntracks-kotlin>



Location search



Reverse Geocoder

- ▶ Input: Latitude, Longitude
- ▶ Output: readable representation (City)

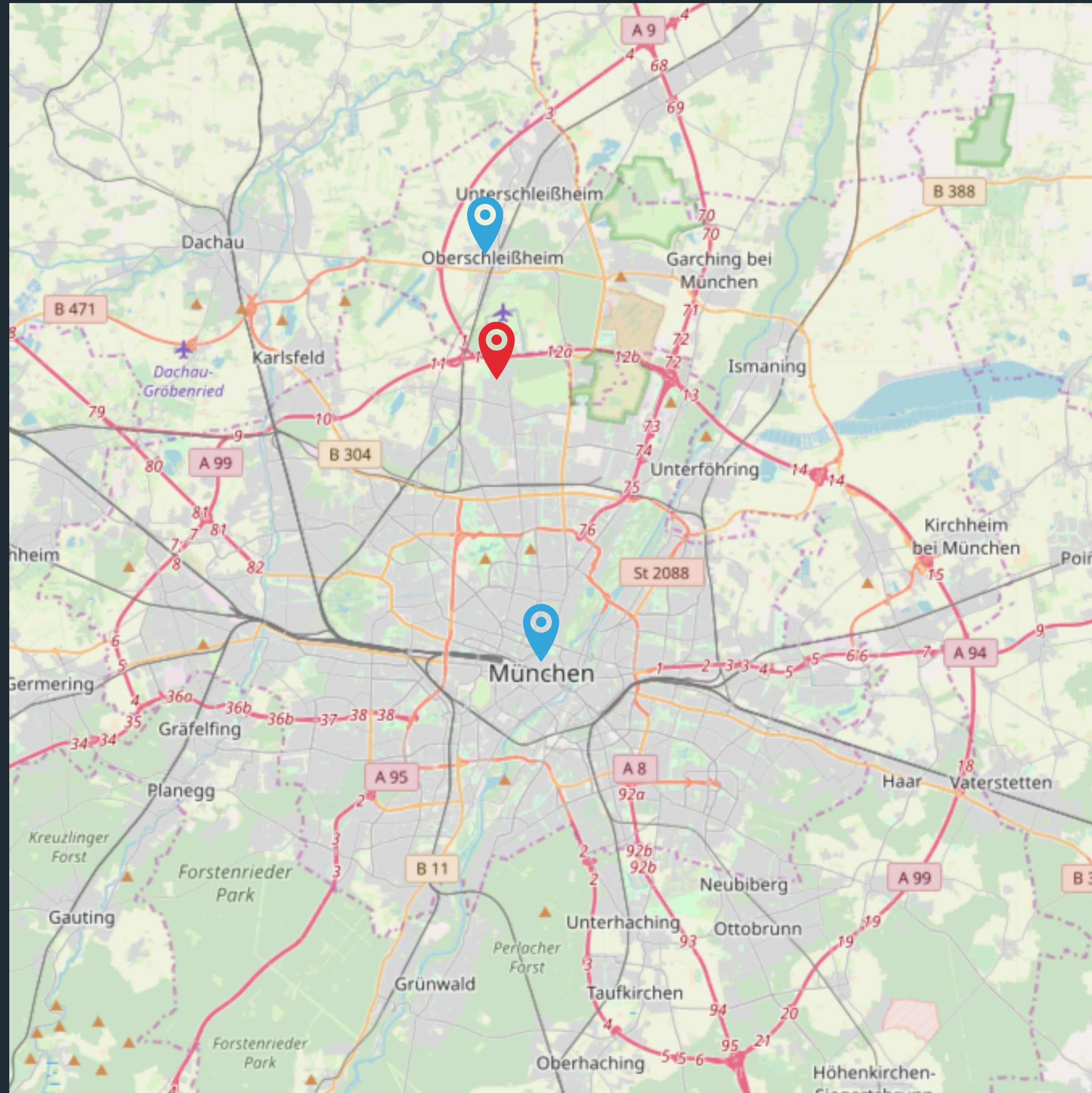


Search across points

- ▶ Each city gets indexed with a lat/lon pair
- ▶ Search for the next point to the supplied one
- ▶ Problem: Neighbours!

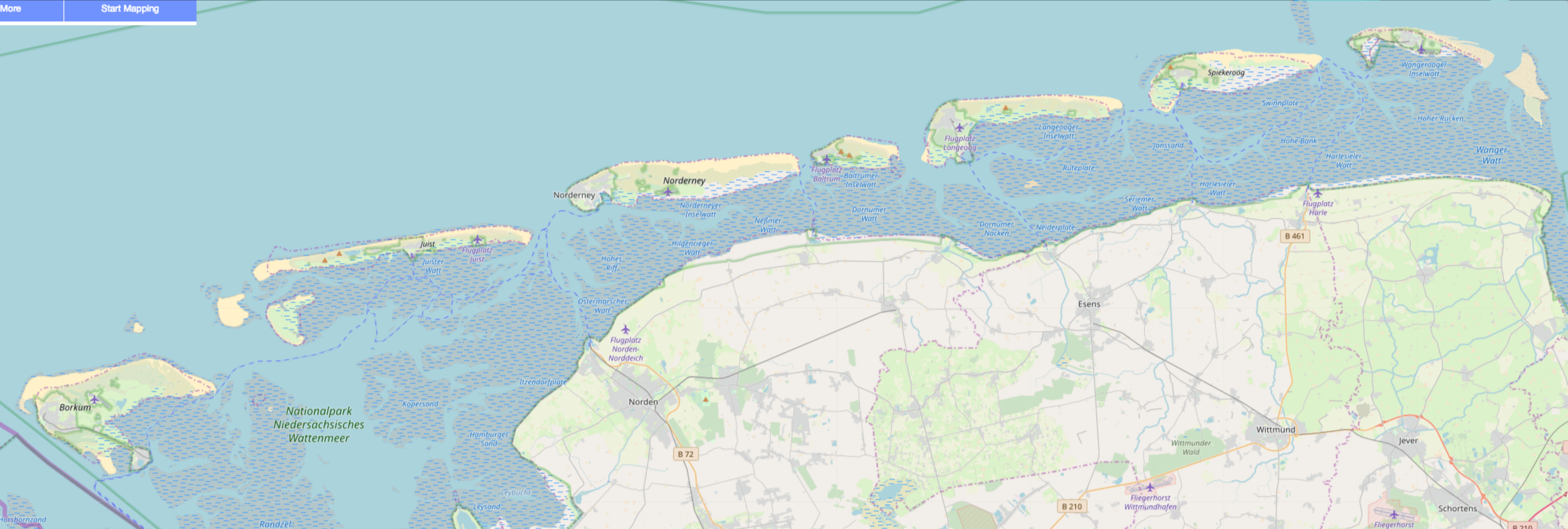


Point based search: Near neighbours



Point based search: Near neighbours

More Start Mapping



Search across shapes

- ▶ Each city gets indexed with a lat/lon pair
- ▶ Certain cities get indexed as a geoshape
- ▶ Run two searches:
 - ▶ Lat/Lon within any shape
 - ▶ Lat/Lon nearby any point

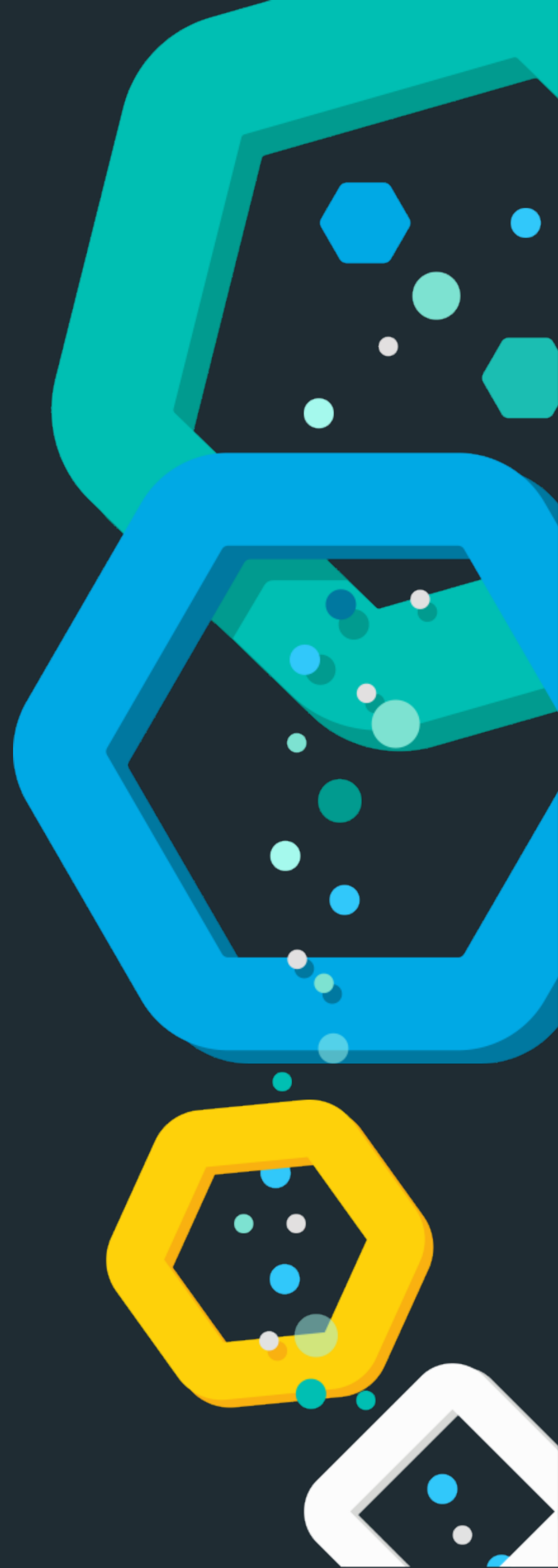


Geo and Lucene: BFF!

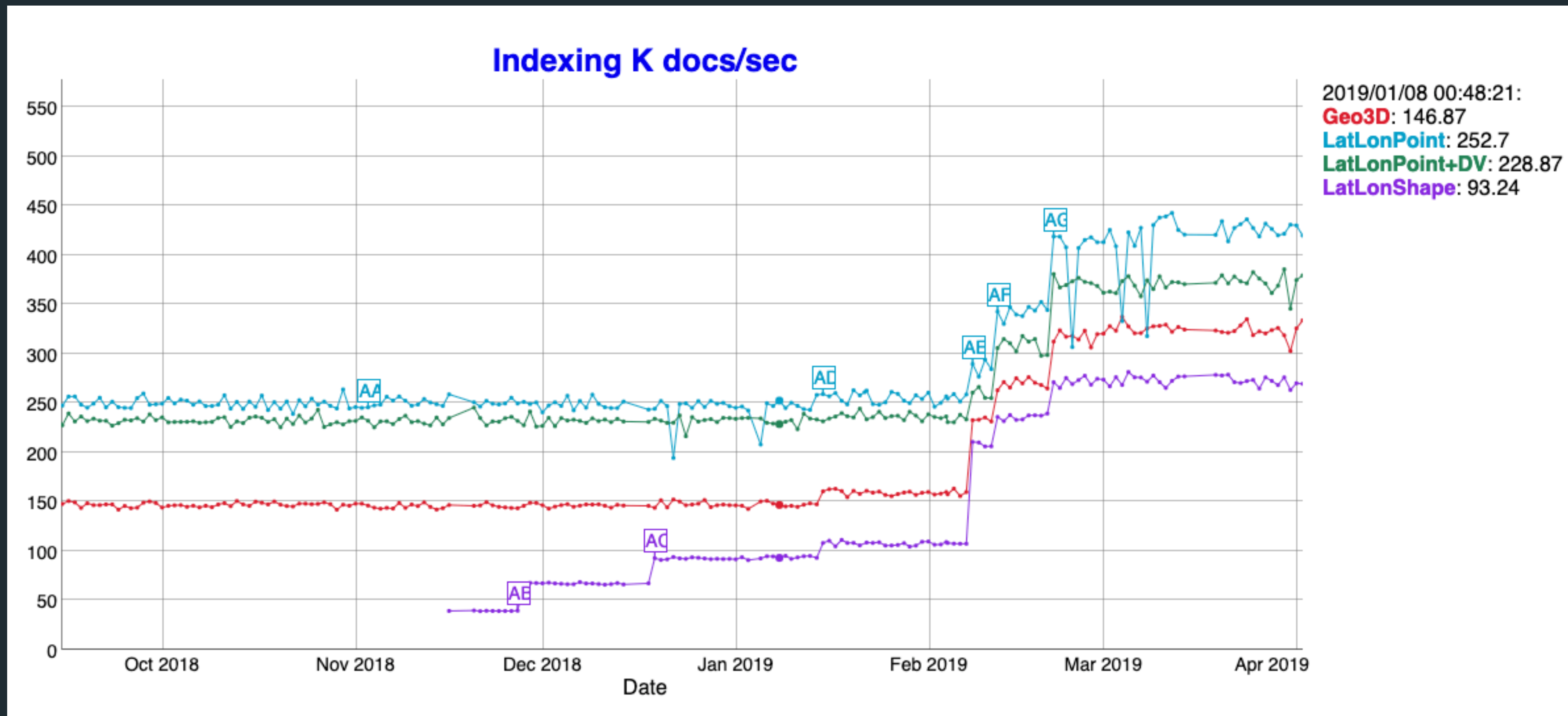
- ▶ **LatLonPoint**: two points, 4 bytes each
- ▶ **LatLonShape**: triangular mesh tessellation

Indexing approach

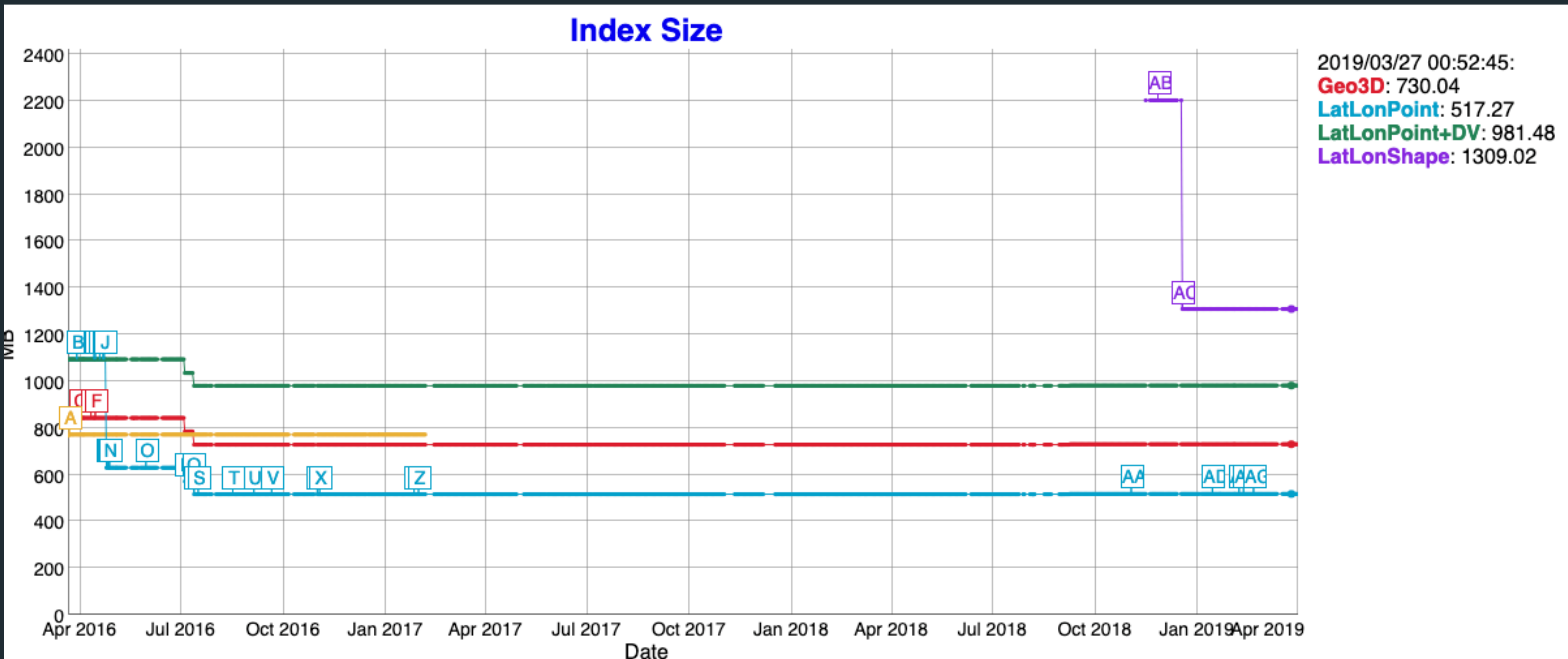
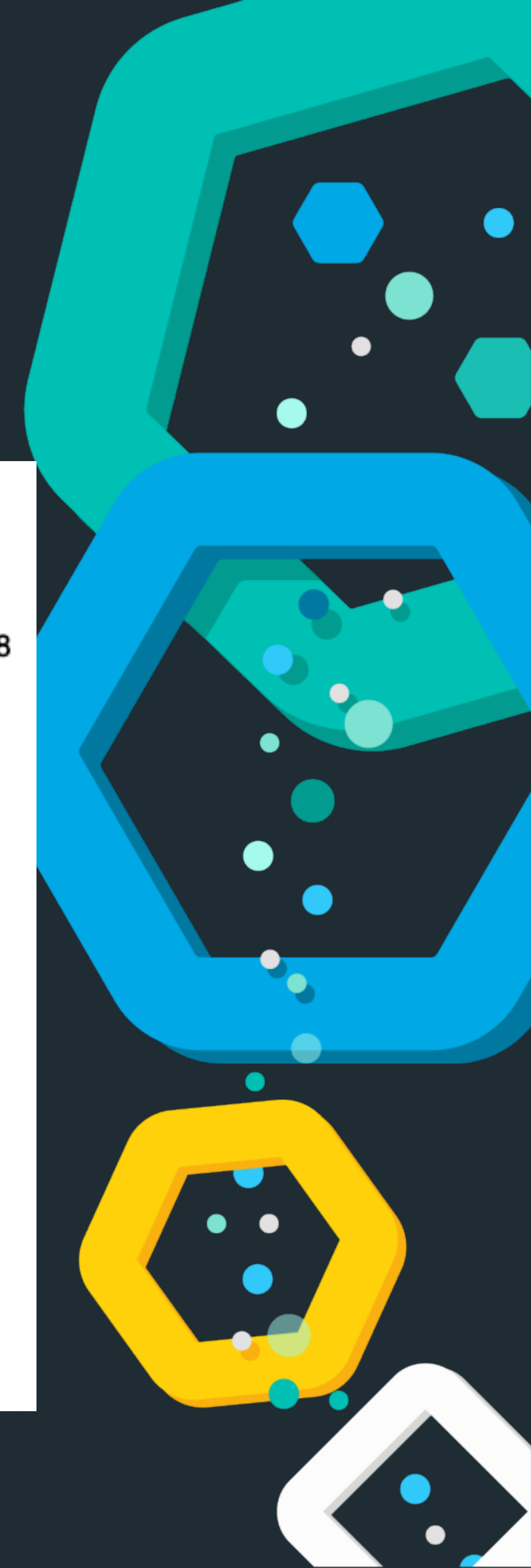
GeoShape types are indexed by decomposing the shape into a triangular mesh and indexing each triangle as a 7 dimension point in a BKD tree. This provides near perfect spatial resolution (down to 1e-7 decimal degree precision) since all spatial relations are computed using an encoded vector representation of the original shape instead of a raster-grid representation as used by the [Prefix trees](#) indexing approach. Performance of the tessellator primarily depends on the number of vertices that define the polygon/multi-polygon. While this is the default indexing technique prefix trees can still be used by setting the `tree` or `strategy` parameters according to the appropriate [Mapping Options](#). Note that these parameters are now deprecated and will be removed in a future version.



Geo and Lucene: BFF!



Geo and Lucene: BFF!

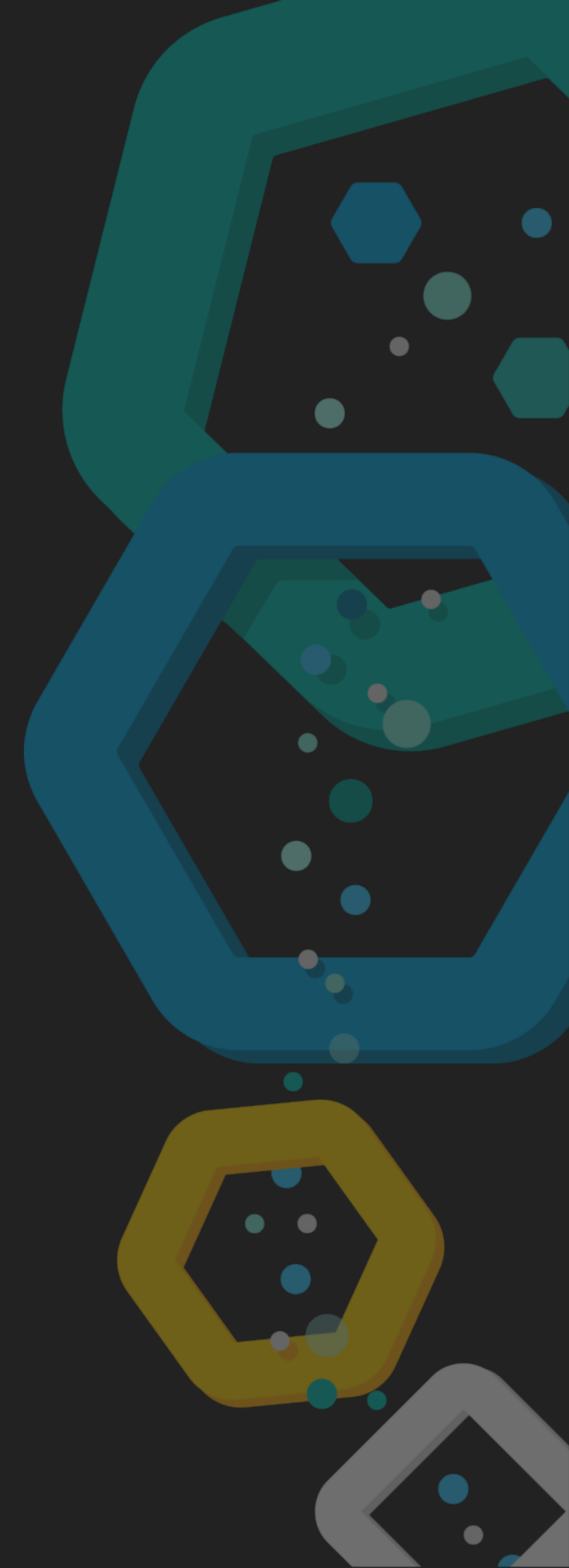


Serverless Lucene

- ▶ Local execution, index part of the package
- ▶ Offline index creation
- ▶ Packaging index into code
- ▶ Index needs to be unpacked, using Lucene via classpath resources is tricky



Demo



Summary

- ▶ Works!
- ▶ Problem: Data quality, getting accurate shape data
- ▶ Problem: First invocation (up to 2s)
 - ▶ JVM startup
 - ▶ Lucene index opening



Faster startup & runtime

Enter GraalVM!

- ▶ A new compiler, supporting HotSpot and AOT compilation
- ▶ Graal compiler part of Java9 (experimental!)
- ▶ Graal JIT compiler part of Java10 (Linux 64bit only)
- ▶ Project Metropolis: Java-on-Java Hotspot implementation
- ▶ Truffle: Framework to implement other languages on top of graal (jruby replacement)



Enter GraalVM!

- ▶ AOT static compilation + SubstrateVM = executable binaries of java apps
- ▶ Using SubstrateVM
- ▶ Reflection!



```
devel/spinscale/aws-lambda-geohash ▶ master ✖ hyperfine --warmup 3 'java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene/ 48.1374 11.5755' './cli/build/cli cli/build/lucene/ 48.1374 11.5755' --prepare 'sudo purge'
```

```
Benchmark #1: java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene/ 48.1374 11.5755  
Time (mean ± σ): 829.6 ms ± 20.1 ms [User: 604.1 ms, System: 184.4 ms]  
Range (min ... max): 804.1 ms ... 862.1 ms 10 runs
```

```
Benchmark #2: ./cli/build/cli cli/build/lucene/ 48.1374 11.5755
```

```
Time (mean ± σ): 100.0 ms ± 2.3 ms [User: 3.5 ms, System: 25.5 ms]  
Range (min ... max): 96.1 ms ... 103.9 ms 11 runs
```

Summary

```
'./cli/build/cli cli/build/lucene/ 48.1374 11.5755' ran  
8.29 ± 0.27 times faster than 'java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene 48.1374 11.5755'
```

```
devel/spinscale/aws-lambda-geohash ▶ master ✖ hyperfine --warmup 3 'java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene/ 48.1374 11.5755' './cli/build/cli cli/build/lucene/ 48.1374 11.5755'
```

```
Benchmark #1: java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene/ 48.1374 11.5755  
Time (mean ± σ): 291.3 ms ± 3.8 ms [User: 549.9 ms, System: 53.7 ms]  
Range (min ... max): 287.4 ms ... 300.0 ms 10 runs
```

```
Benchmark #2: ./cli/build/cli cli/build/lucene/ 48.1374 11.5755
```

```
Time (mean ± σ): 4.9 ms ± 0.6 ms [User: 2.3 ms, System: 1.5 ms]  
Range (min ... max): 4.3 ms ... 7.6 ms 425 runs
```

Warning: Command took less than 5 ms to complete. Results might be inaccurate.

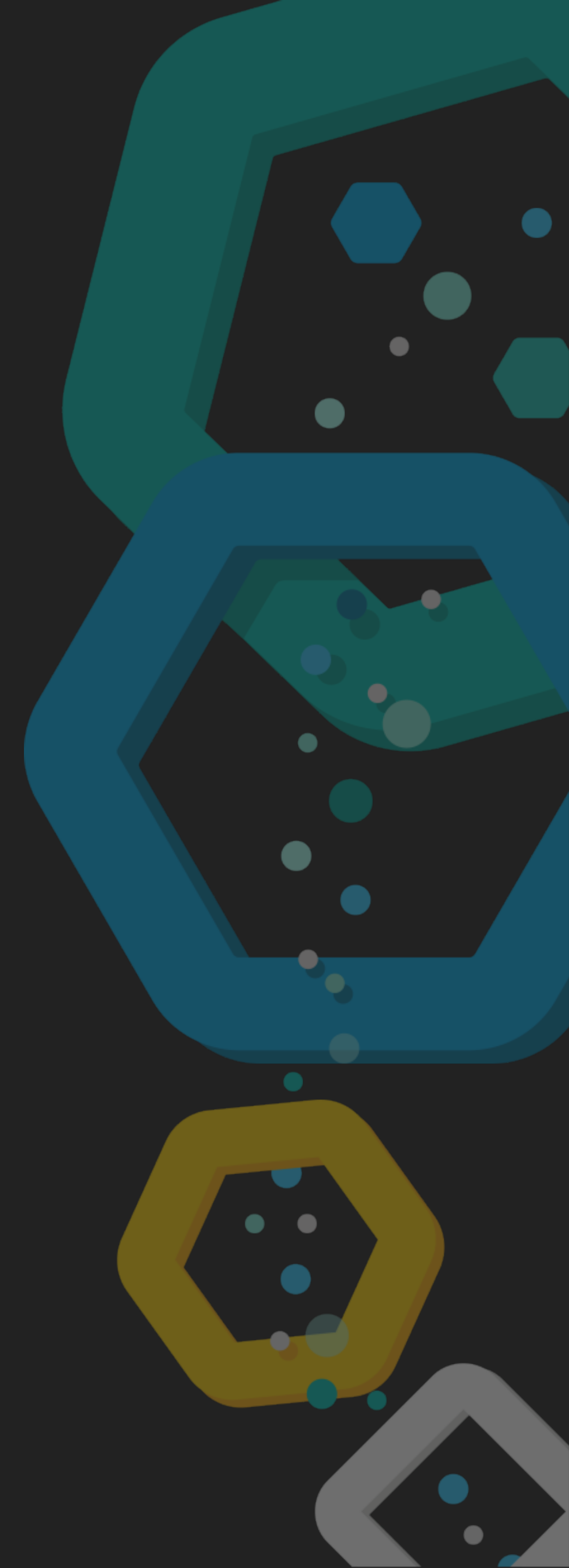
Summary

```
'./cli/build/cli cli/build/lucene/ 48.1374 11.5755' ran  
59.01 ± 7.27 times faster than 'java -jar cli/build/libs/cli-all-0.1.jar cli/build/lucene 48.1374 11.5755'
```

```
devel/spinscale/aws-lambda-geohash ▶ master ✖
```



Deployment model

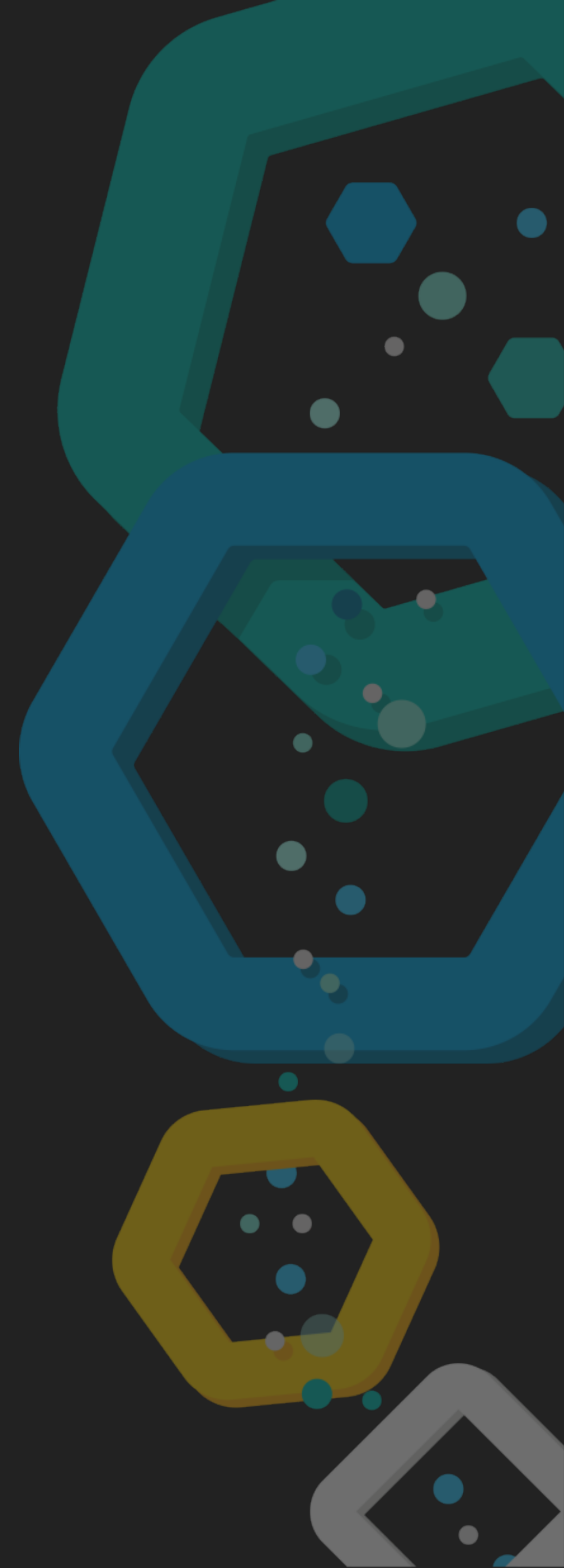


AWS Lambda

- ▶ Deployment model: Zip archive in S3 bucket
- ▶ Execution: Download zip archive & execute
- ▶ Requires regular java start (AWS reduced JVM startup time)
- ▶ GraalVM can only be used with custom runtime



runtime flow



**CUSTOM
RUNTIME**

```
AWS_LAMBDA_RUNTIME_API=localhost:12345  
_HANDLER="my_handler"
```

```
/bin/bootstrap
```



```
AWS_LAMBDA_RUNTIME_API=localhost:12345  
_HANDLER="my_handler"
```

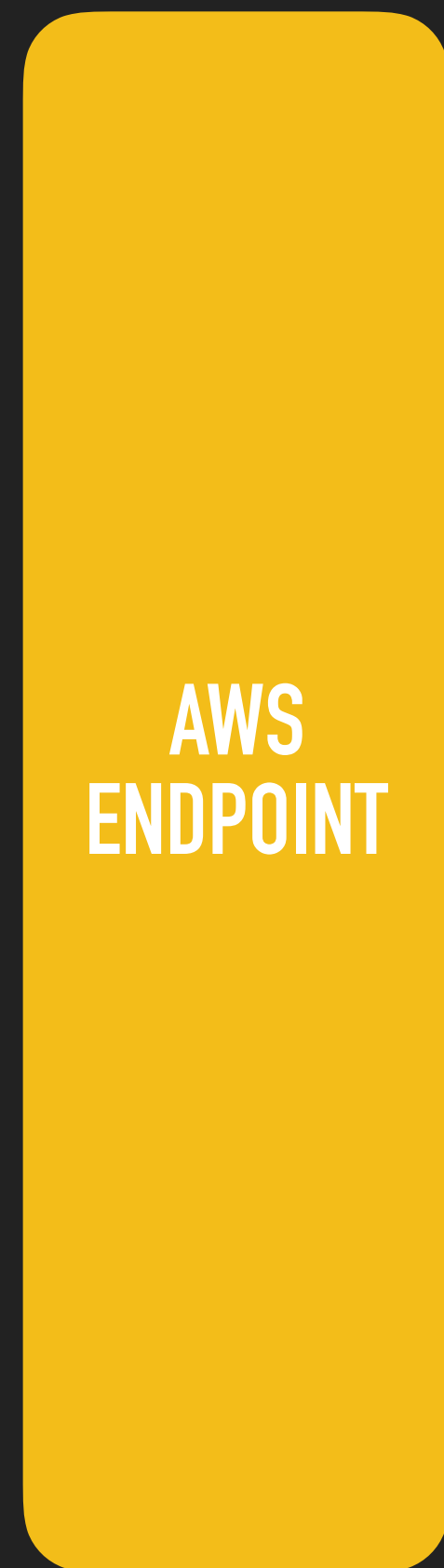
```
/bin/bootstrap
```

AWS
ENDPOINT

CUSTOM
RUNTIME

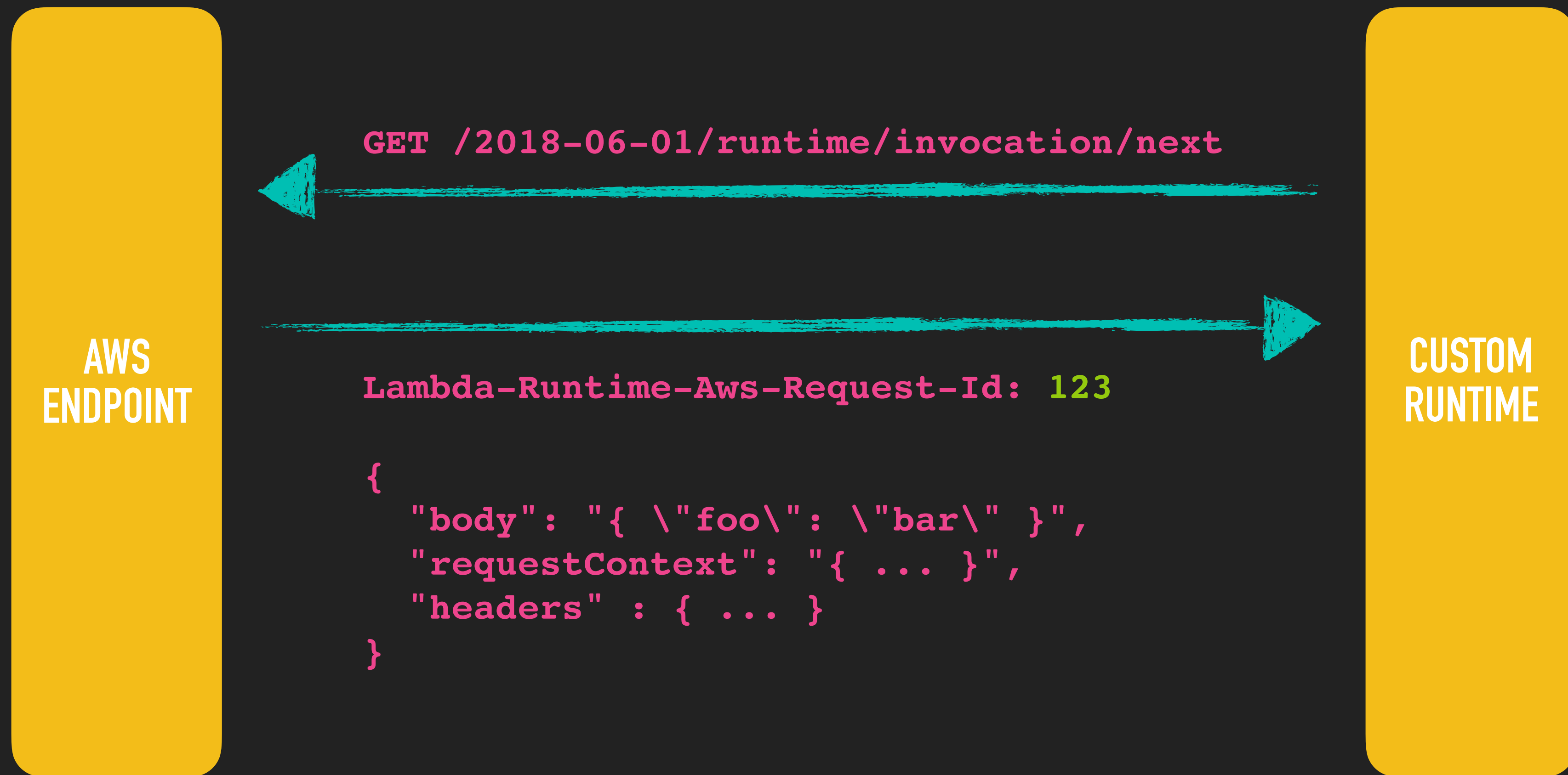

```
AWS_LAMBDA_RUNTIME_API=localhost:12345  
_HANDLER="my_handler"
```

```
/bin/bootstrap
```



```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"
```

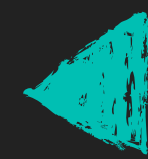
```
/bin/bootstrap
```



```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"
```

```
/bin/bootstrap
```

```
POST /2018-06-01/runtime/invocation/123/response
```



```
{
  "body": "{ \"foo\": \"bar\" }",
  "headers": { ... }
}
```

AWS
ENDPOINT

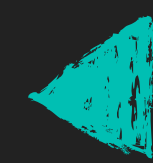
CUSTOM
RUNTIME

```
AWS_LAMBDA_RUNTIME_API=localhost:12345
_HANDLER="my_handler"
```

```
/bin/bootstrap
```

AWS
ENDPOINT

```
POST /2018-06-01/runtime/invocation/123/response
```



```
{
  "body": "{ \"foo\": \"bar\" }",
  "headers": { ... }
}
```

```
HTTP OK 202
```



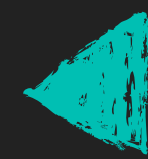
CUSTOM
RUNTIME

```
AWS_LAMBDA_RUNTIME_API=localhost:12345  
_HANDLER="my_handler"
```

```
/bin/bootstrap
```

AWS
ENDPOINT

```
POST /2018-06-01/runtime/invocation/123/error
```



```
{  
  "statusCode": 500,  
  "body" : "..."  
}
```

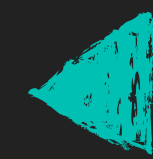
CUSTOM
RUNTIME

```
AWS_LAMBDA_RUNTIME_API=localhost:12345  
_HANDLER="my_handler"
```

```
/bin/bootstrap
```

AWS
ENDPOINT

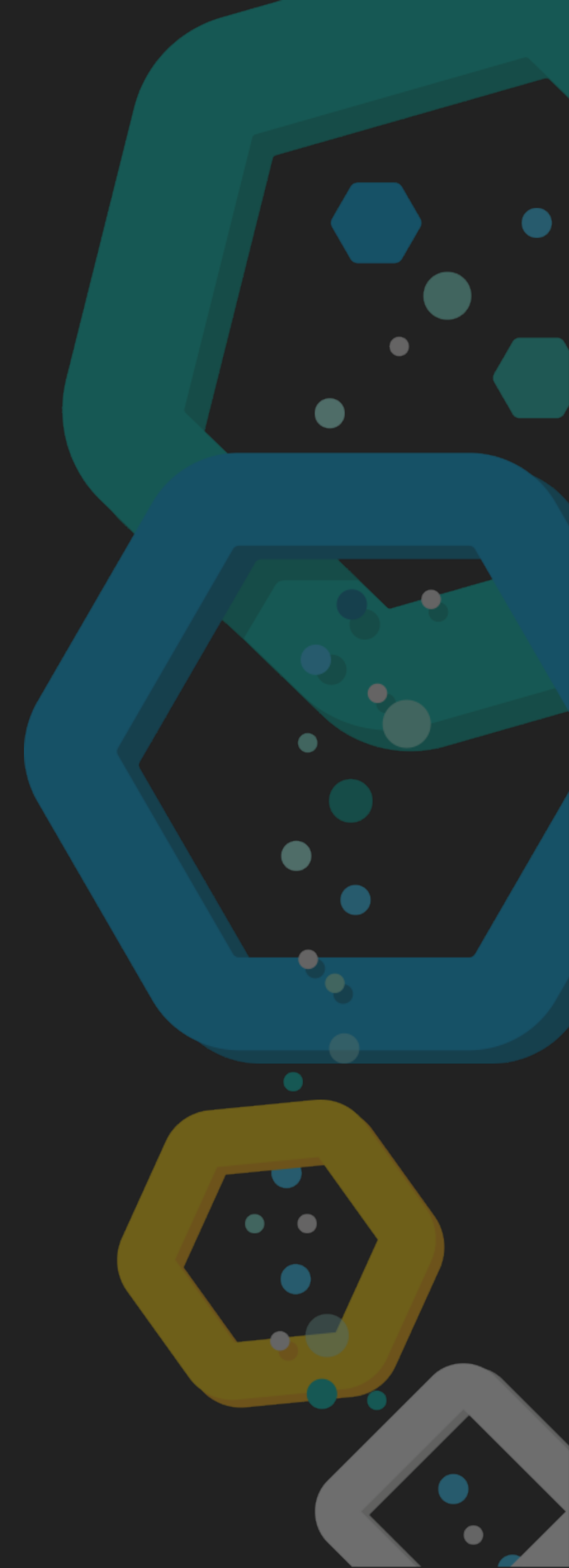
POST /2018-06-01/runtime/init/error



```
{  
  "errorMessage" : "...",  
  "errorType" : "..."  
}
```

CUSTOM
RUNTIME

Google Cloud Run

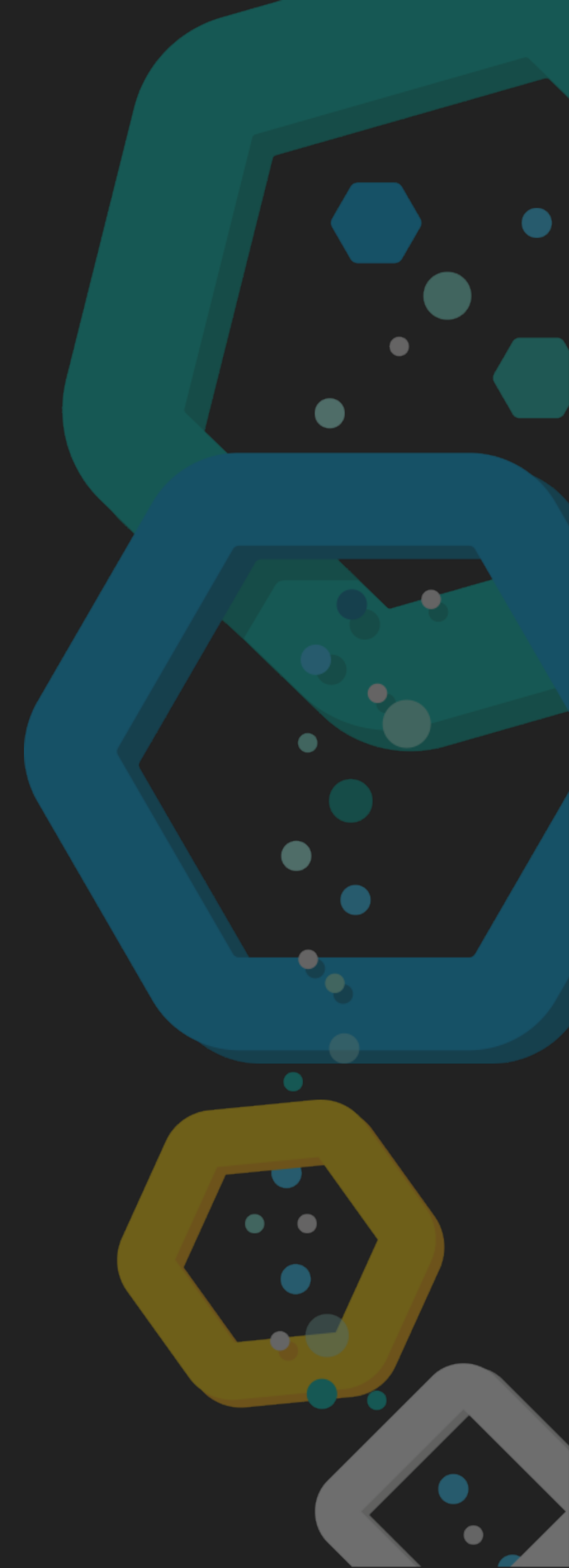


Google Cloud Run

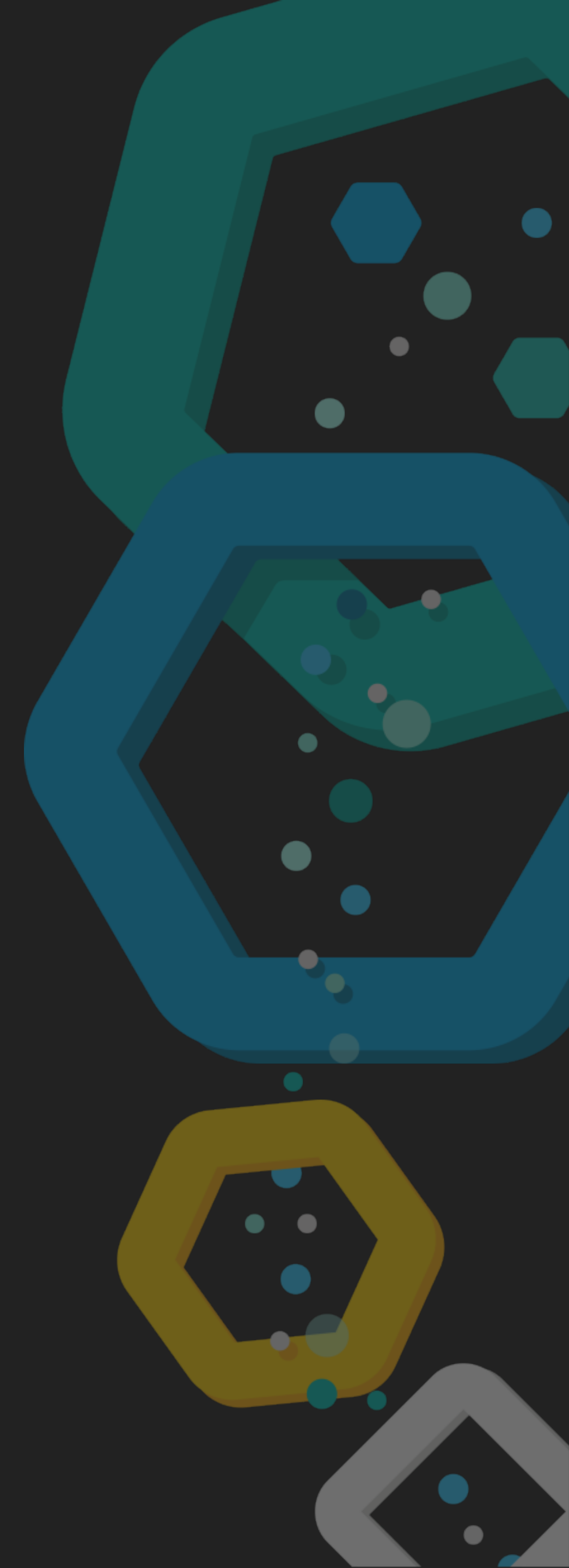
- ▶ Serverless done 'right'?
- ▶ Docker container as a web application listening in \$PORT
- ▶ Deployment model: **docker push && gcloud beta run**
- ▶ Easier to test
- ▶ Configurable concurrency
- ▶ Improved billing model due to concurrency



Demo

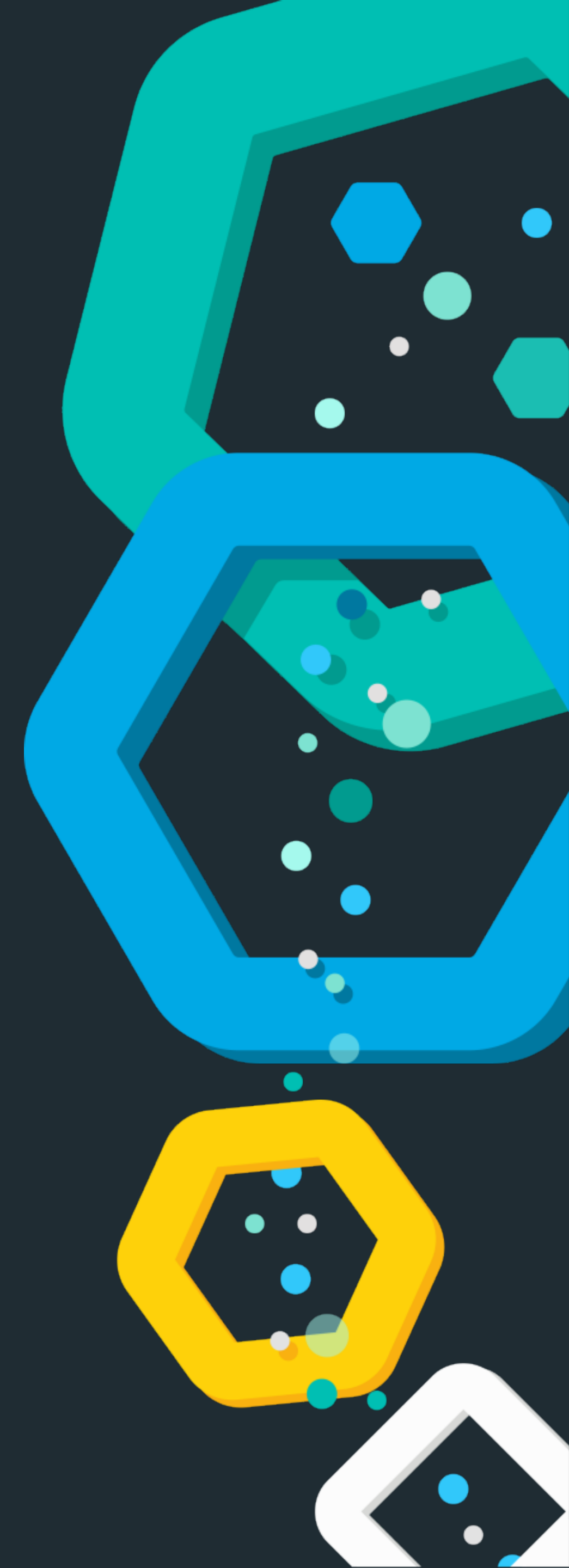


Summary



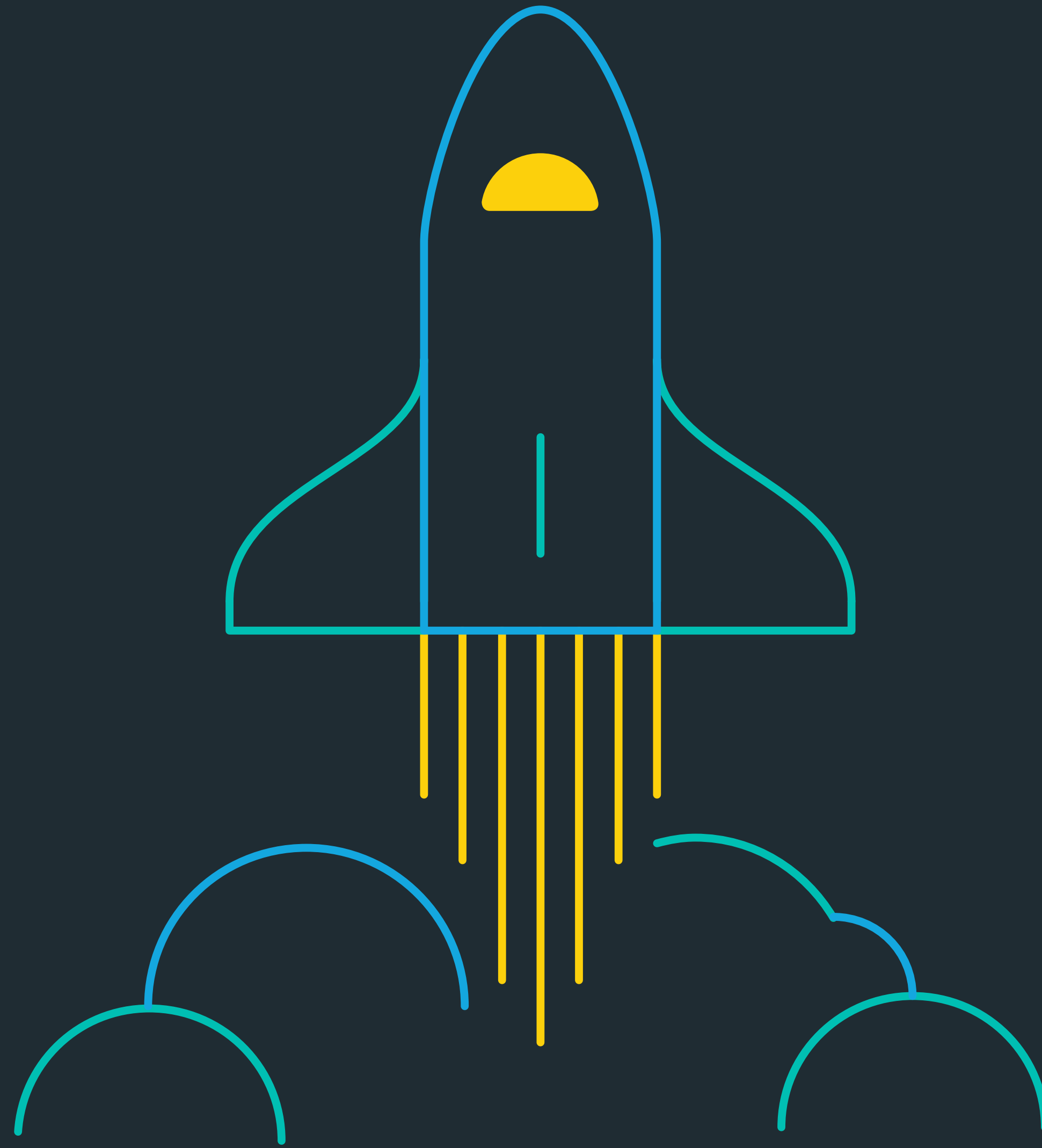
Summary

- ▶ Economics
 - ▶ Billing model
 - ▶ Runtime cost vs. development cost
 - ▶ Break even vs. non serverless model
- ▶ Development
 - ▶ Vendor lock-in
 - ▶ Deployment model is important
 - ▶ Cloud Run: Run serverless or as container
 - ▶ Scalability strategies, base load container, increased load serverless?
 - ▶ Observability



Discussion

... ask all the things!



Links

- ▶ <https://serverless.com/framework/docs/>
- ▶ <https://www.openfaas.com>
- ▶ <https://cloud.google.com/knative/>
- ▶ <https://kubernetes.io/>
- ▶ <https://fission.io/>
- ▶ <http://fnproject.io/>
- ▶ <https://nuclio.io>
- ▶ <https://openwhisk.incubator.apache.org/>

- ▶ <https://www.graalvm.org>
- ▶ <https://openjdk.java.net/projects/metropolis/>
- ▶ <https://github.com/oracle/graal/tree/master/substratevm>

- ▶ https://en.wikipedia.org/wiki/Reverse_geocoding

