



SOUNDCLOUD

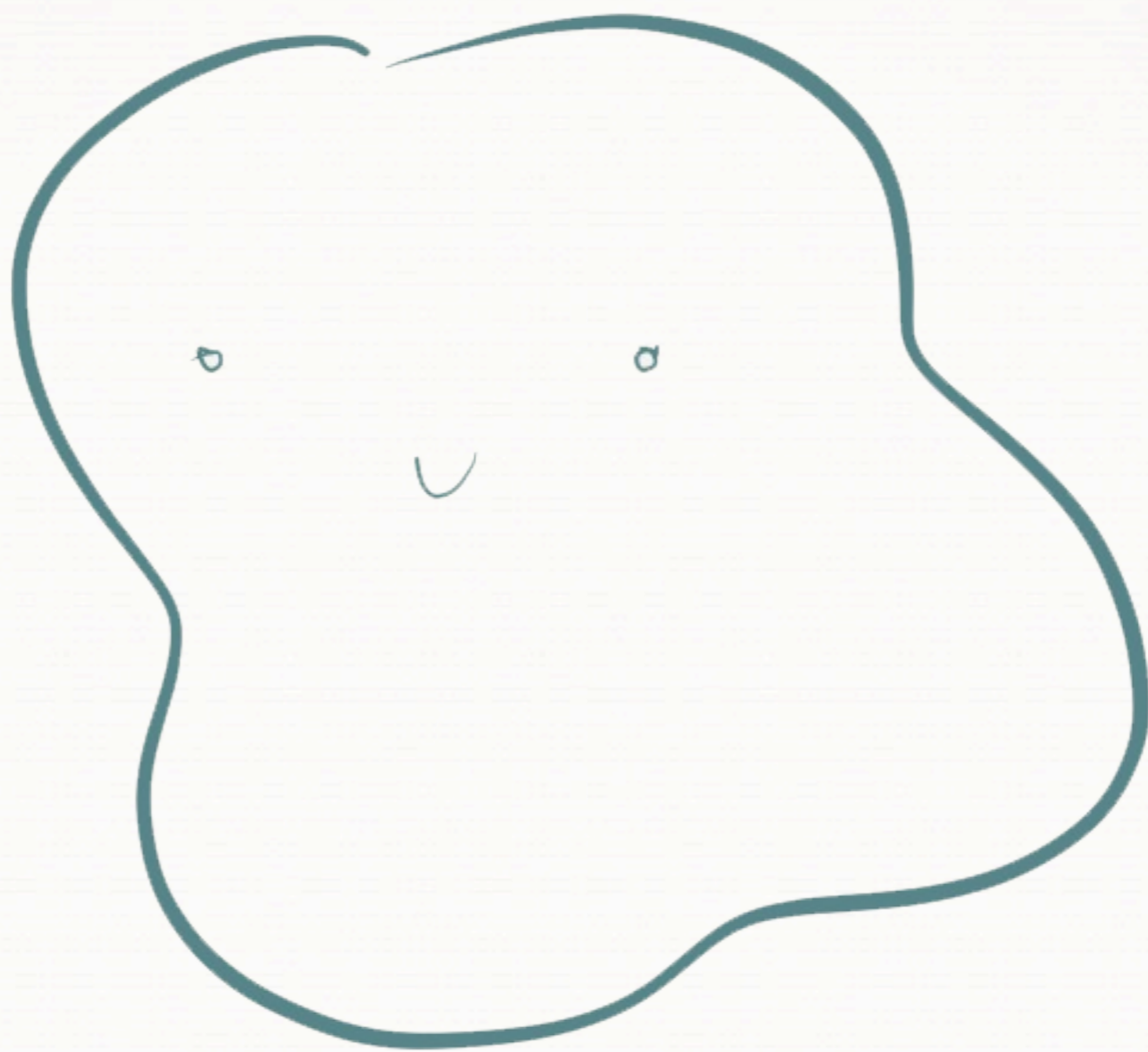
ElasticSearch and SOA

Peter Bourgon

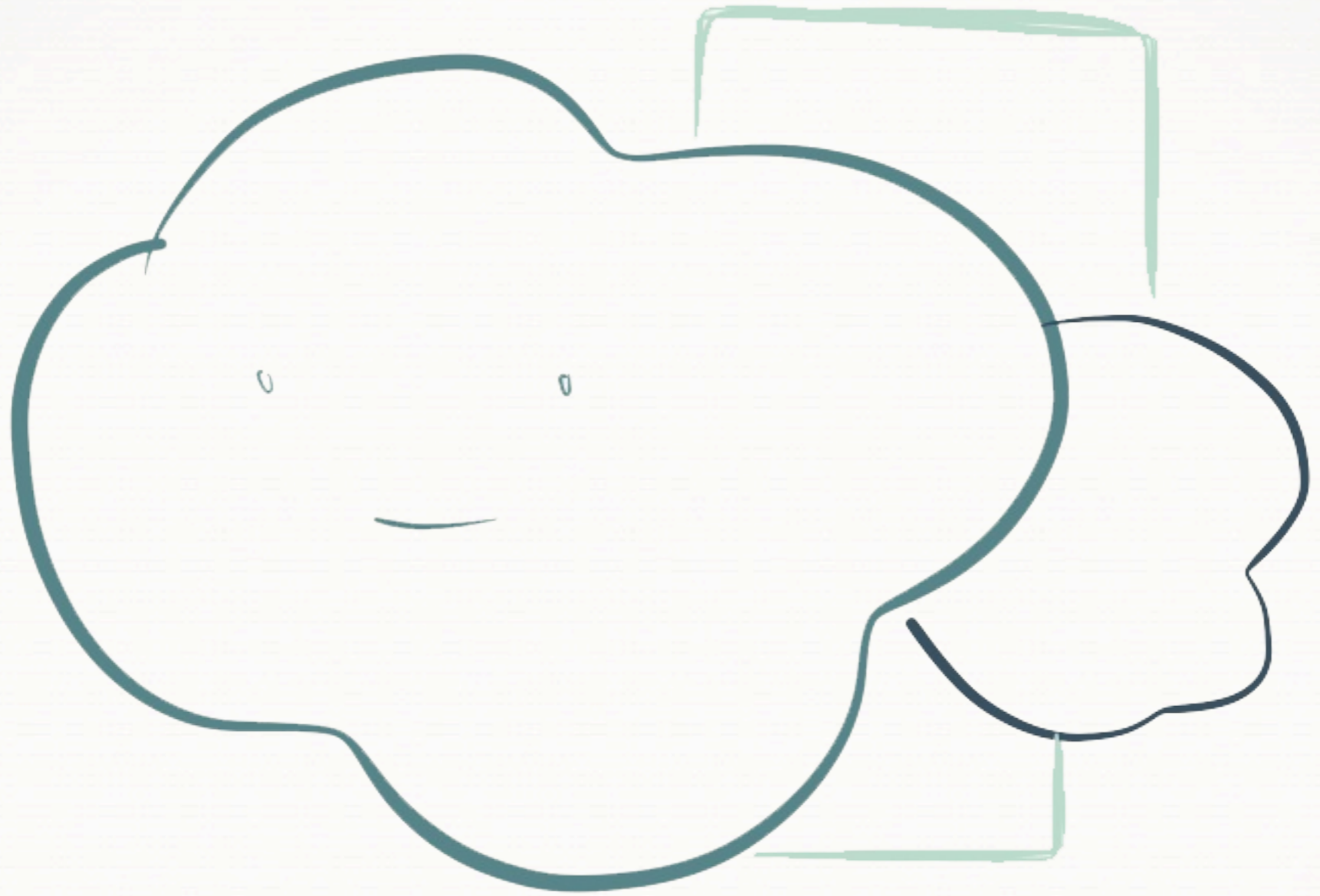
Evolution of search
Current architecture
A bit about SOA

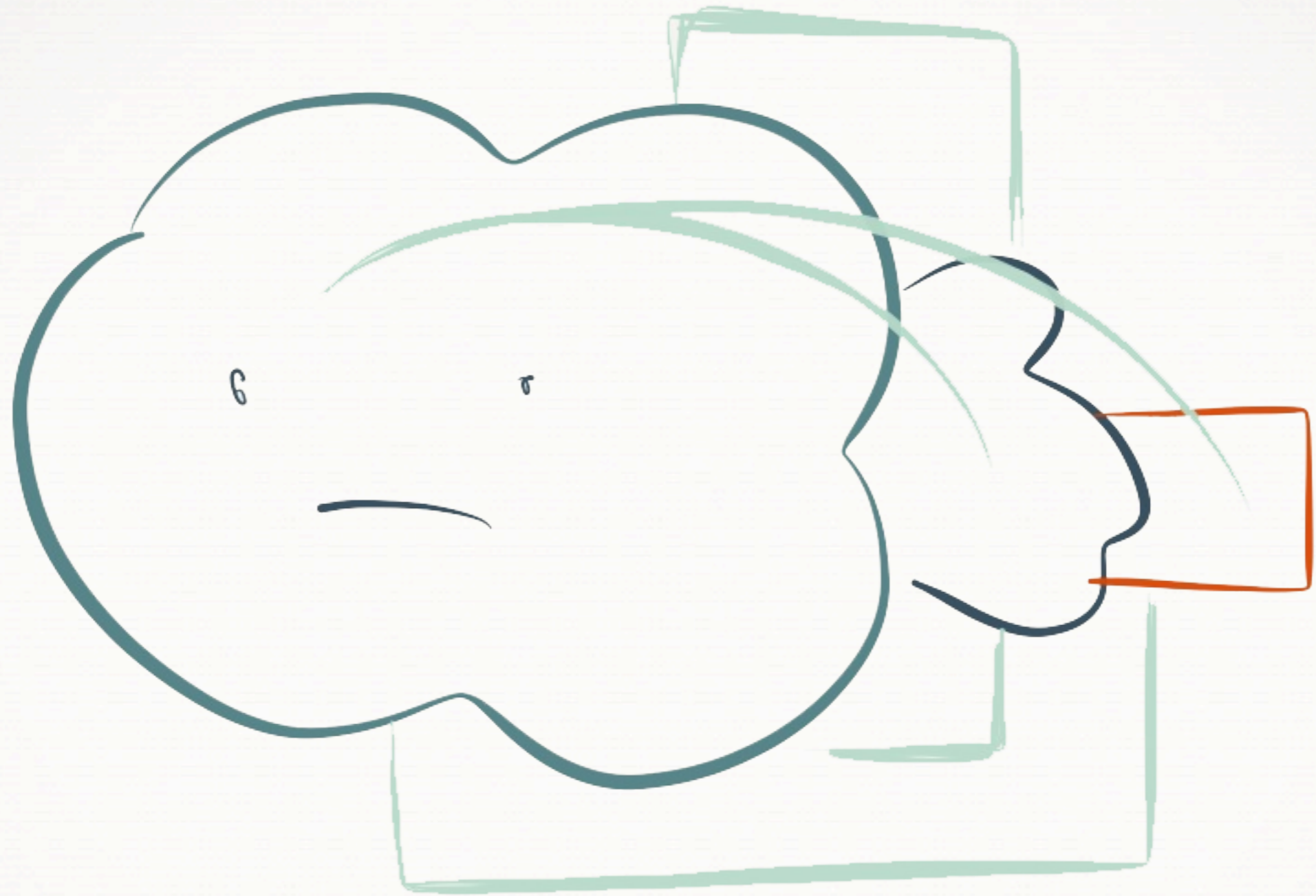


EVOLUTION OF SEARCH











KATAMARI DAMACY, NAMCO



:(

- ✘ Solr**
- ✘ Batch replication**
- ✘ Very slow reindexing**
- ✘ Technical debt in app**

DO WANT.

- ✓ Velocity
- ✓ Reliability
- ✓ Maintainability



HOW TO DO?

ElasticSearch

Greenfield development

Parallel dark launch

Migrate traffic slowly

Scale out



WHY ELASTICSEARCH

- ✓ Good API
- ✓ Clear path for growth
- ✓ Batteries included
- ✓ Works like you expect it to



OUR ELASTICSEARCH

External versioning

No dynamic mapping

Multiple indices ('live' alias)

Custom scoring module

Index: $2N$ shards, 1 replica

Cluster per datacenter



IMPORTANT SETTINGS

discovery:

zen:

minimum_master_nodes: $(N/2)+1$



IMPORTANT SETTINGS

cluster:

routing:

allocation:

node_concurrent_recoveries: 1



IMPORTANT SETTINGS

bootstrap:
mlockall: true

-Xmx = -Xms = (memory/2)



IMPORTANT LESSON

One cluster = single point of failure

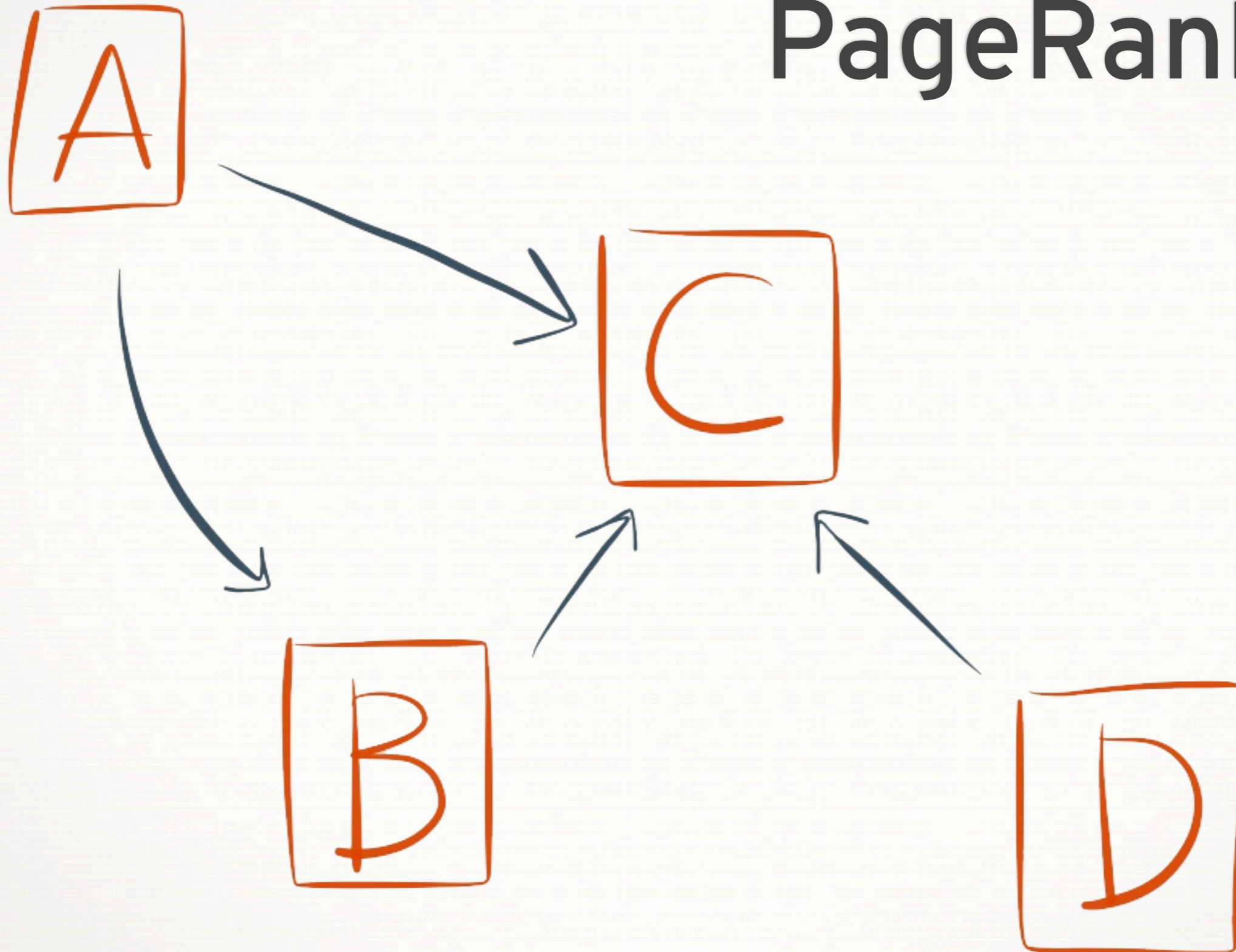


OUR SEARCH COMPONENTS

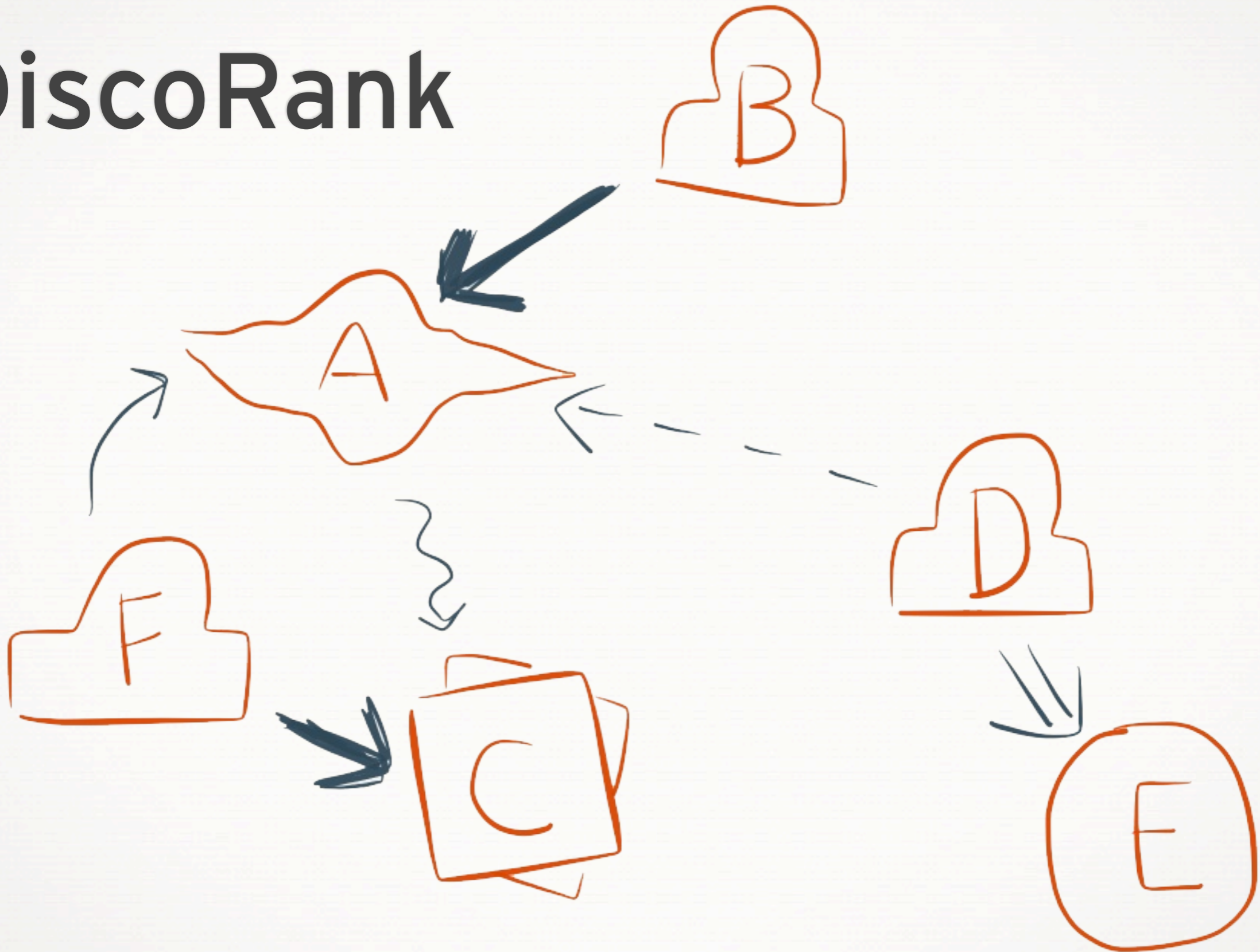


DISCORANK

PageRank



DiscoRank



DISCORANK STRATEGY

Calculations done offline
Serialize to compact array
Load in ES custom scorer
Recalculate/reload every N



INDEXING

Single-purpose application

Stateless and idempotent

Full catalog \approx 1 hour

Build, iterate, iterate, iterate



SEARCHING

Single-purpose application

Stateless and idempotent

SC/ES DSL translation

Open-source ES library



SUGGESTING

Single-purpose application

“Stateless” and idempotent

Prefix trie in memory

Lucene Finite State Transducer



SERVICE ORIENTED ARCHITECTURE

SOA AT SOUNDCLLOUD

Bazooka platform

API as firewall

12 Factor applications



12 FACTOR APPS

Single codebase (repository)

Config stored in environment

Backing services as resources

Stateless

Horizontal scaling with processes



WHY SOA?

Velocity

Developer happiness

Distributed systems are complex



SOA infrastructure

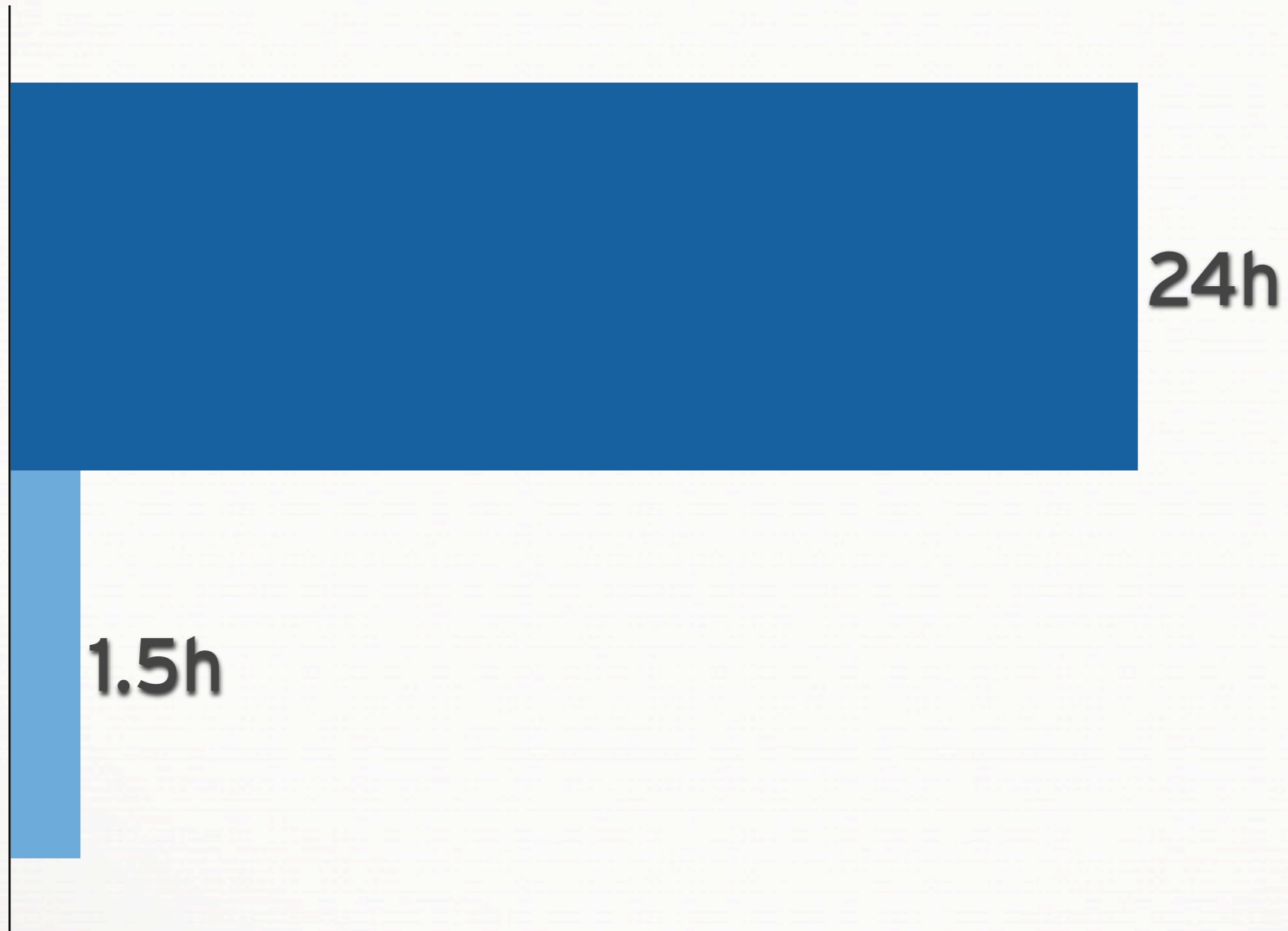
ElasticSearch core



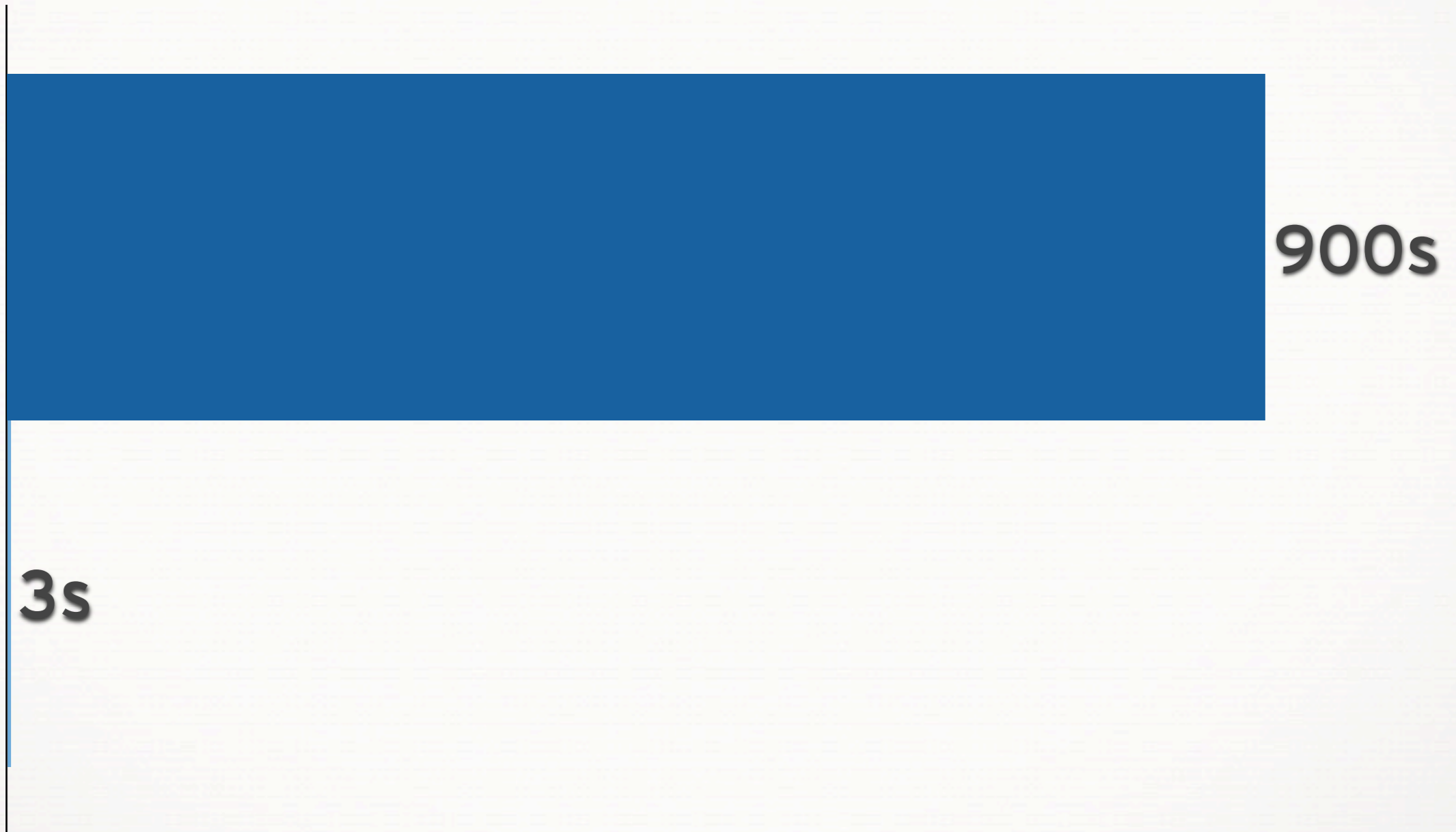
QUICK STATS



TIME TO REINDEX



TIME TO VISIBILITY



OTHER STATS

Cleared (big!) feature backlog

> 2x traffic since project start

> 3x corpus

Flat response curve



WIN

ElasticSearch works as expected
SOA = force multiplier



Thanks!

Questions?

Peter Bourgon
peter.bourgon.org
@peterbourgon

