
AWS
re:Invent

ADM301

Running ultra-high throughput ad tech workloads on Aerospike

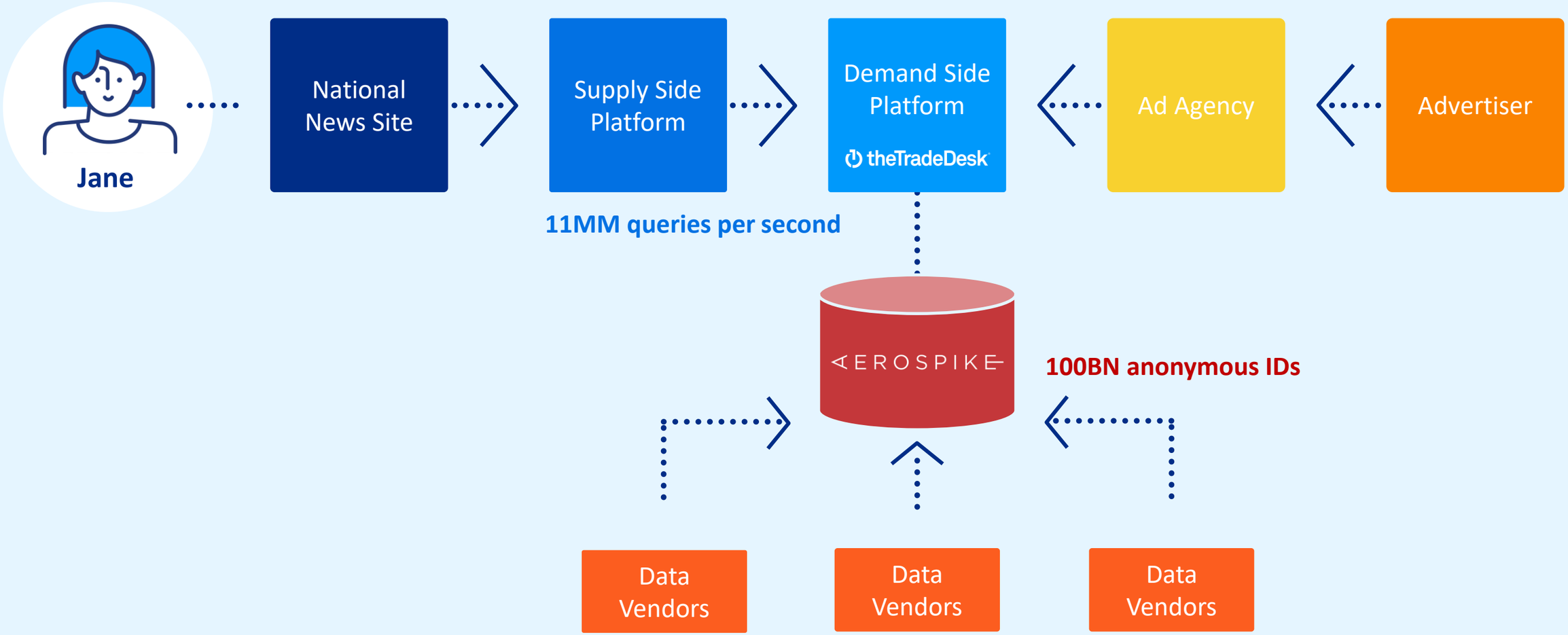
Gerry Louw
AWS

Matt Cochran
Director of Engineering
The Trade Desk

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction to The Trade Desk



Aerospike and The Trade Desk

- The Trade Desk has been a customer for almost 10 years
- Run some of the largest clusters in the world
- Aerospike is the best no-SQL solution to support our millions of QPS workloads
- Amazingly simple to operate; dynamic cluster management and scalability is key



theTradeDesk®

AWS and The Trade Desk

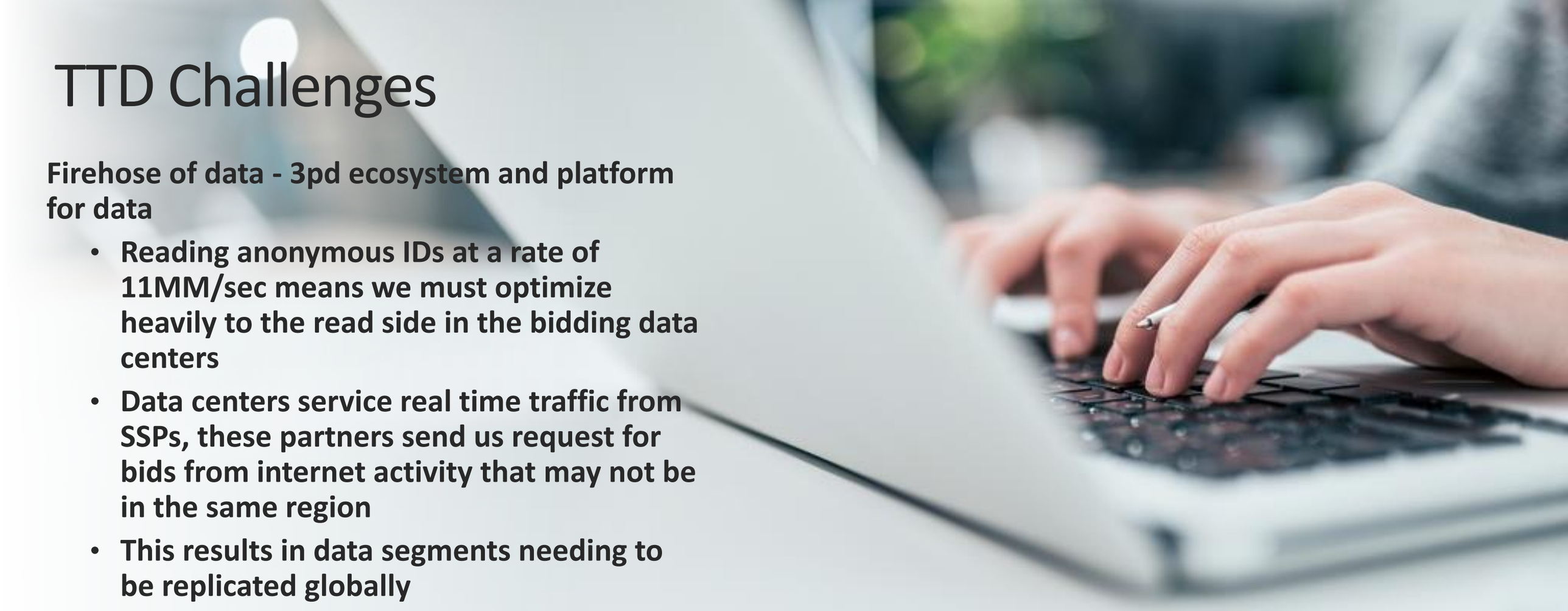
- The Trade Desk has been a customer for over 8 years
- Run multiple workloads; from backend processing to data warehousing to EMR
- Scalability and elasticity have been critical to success



TTD Challenges

Firehose of data - 3pd ecosystem and platform for data

- **Reading anonymous IDs at a rate of 11MM/sec means we must optimize heavily to the read side in the bidding data centers**
- **Data centers service real time traffic from SSPs, these partners send us request for bids from internet activity that may not be in the same region**
- **This results in data segments needing to be replicated globally**
- **Serialization/Deserialization CPU costs on the read side mean we have to be careful about TDID record size**
- **100BN records - many with tens of thousands of data points or segments**



TTD Challenges

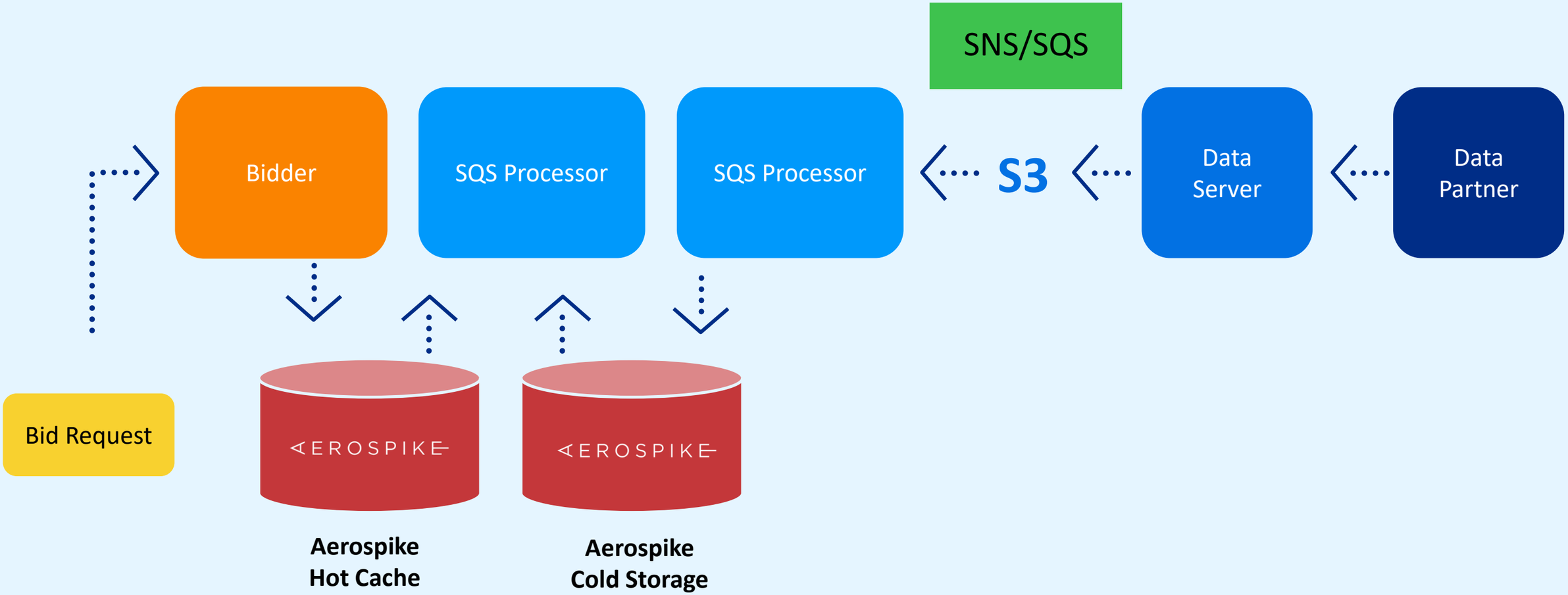
We're a platform for data, so data partners need to send us data in advance and sell it in our marketplace

- **Not all data points are used**
- **Usage comes and goes as ad agency customers start and end campaigns targeting one or more of these segments**
- **3rd party data partners send us over 100K QPS and each query can add one or more segments to one or many TDIDs**
- **This turns into about 30MM TDID/segment updates a second**
- **Being careful means CPU costs for merging data onto TDID records as we have to pick the 'best' elements if a given record gets too large**
- **DCs need only the data on the record that is in use by an active campaign**
- **Duplicative processing and storage – replicating to each DC means we pay the CPU and storage cost each time**

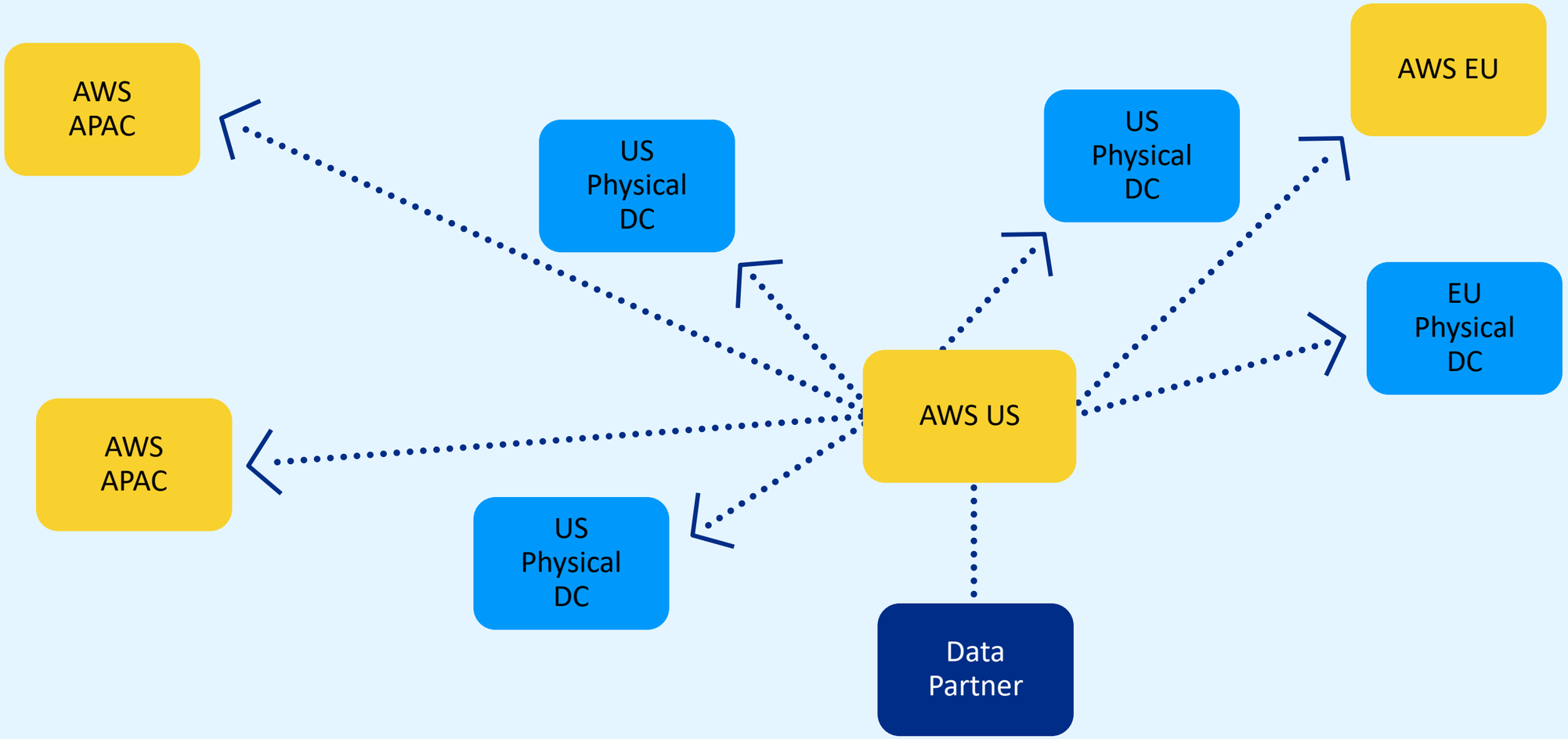


theTradeDesk

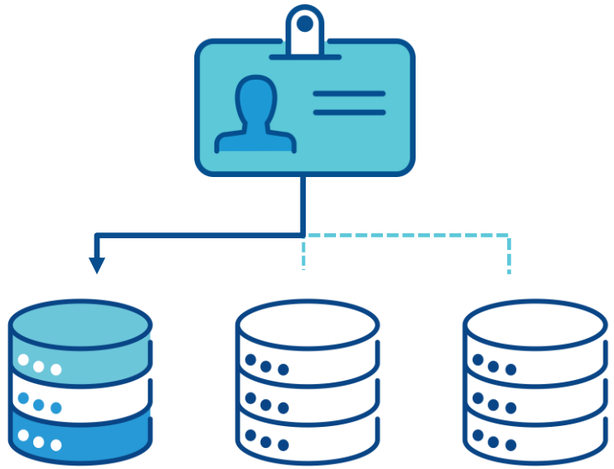
Overview of system flow



Overview of global system flow



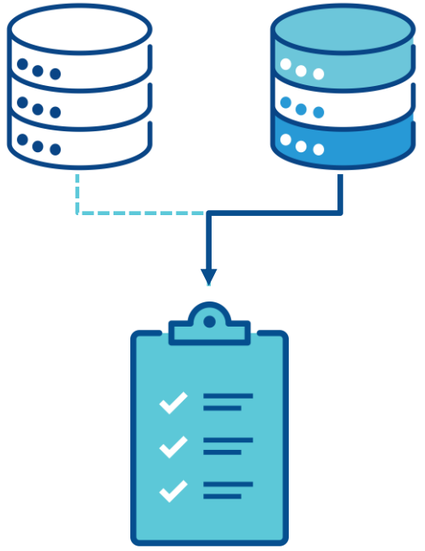
Data storage use cases



Hot Cache TDID case – TDIDs can come and go from data centers

- How to get only the TDIDs in the data center they are needed in
- How to store only the TDIDs that are active
- Classic hot cache concept

Base record:
Anonymous TDID Record
-> list of segments



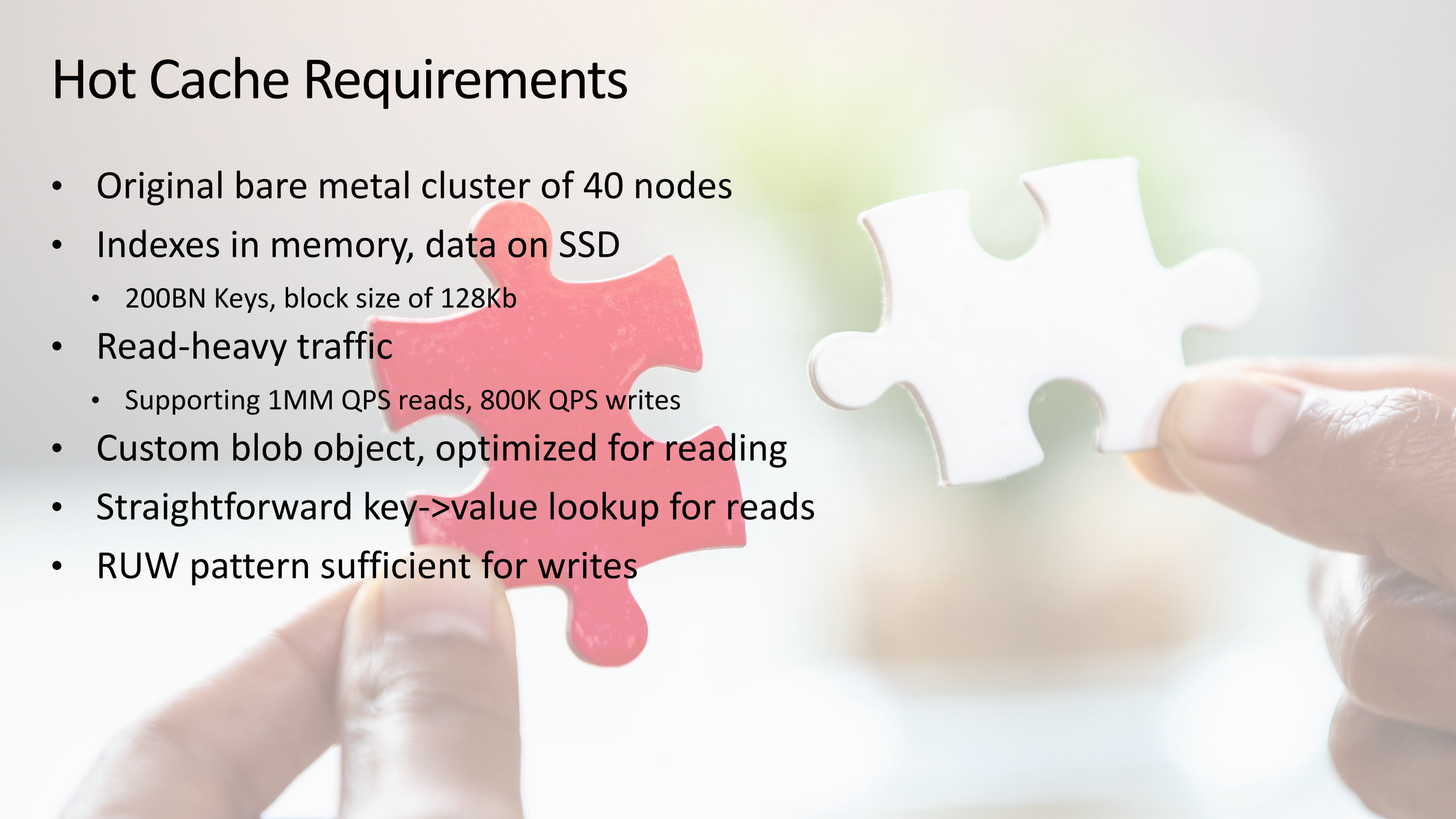
Cold Storage/Reverse index: List of TDIDs for a given segment

- Data usage state change between active/inactive
- How to keep only the actively used segment data onto the record
- How to thaw segments that change to active

theTradeDesk

Hot Cache Requirements

- Original bare metal cluster of 40 nodes
- Indexes in memory, data on SSD
  - 200BN Keys, block size of 128Kb
- Read-heavy traffic
  - Supporting 1MM QPS reads, 800K QPS writes
- Custom blob object, optimized for reading
- Straightforward key->value lookup for reads
- RUW pattern sufficient for writes



Hot cache AWS - Sizing

- Aerospike Cert Tool – use ACT to size, this gives you a clear picture
- Try multiple instances with this – AWS has a fantastic variety of instance types
- Find the balance of CPU/IO/Storage
- Run it for a day to catch issues
- We ended up with 13.ens as the most balanced/cost effective
- i3.ens have so much more storage space but we do sometimes get memory constrained



theTradeDesk

Hot cache AWS Operations

- Use CloudFormation to spin up instead of ASGs, because automatic termination doesn't work for a distributed database
- Chef bootstrapping
- Physical SSDs are 4x faster than NVME, but elastic is a better tradeoff



theTradeDesk®



Cold Storage Requirements

- Write 30MM key/value tuples/sec
- Cold cache to reduce in-dc footprint
- Buffer to absorb data faster
- Reads far lighter
- 1Pb data set uncompressed

theTradeDesk

Cold Storage History

- Began life as Cassandra, not Aerospike
- Compression required more CPU
- GC also killed us on CPU
- Grew to > 500 i3.2xl nodes
- High amount of CPU for the amount of data stored
- Data model doesn’t work for both fast writes and reads
- Aerospike solves for GC and efficient data model
- Amazon EC2 solves for initial POC and growth



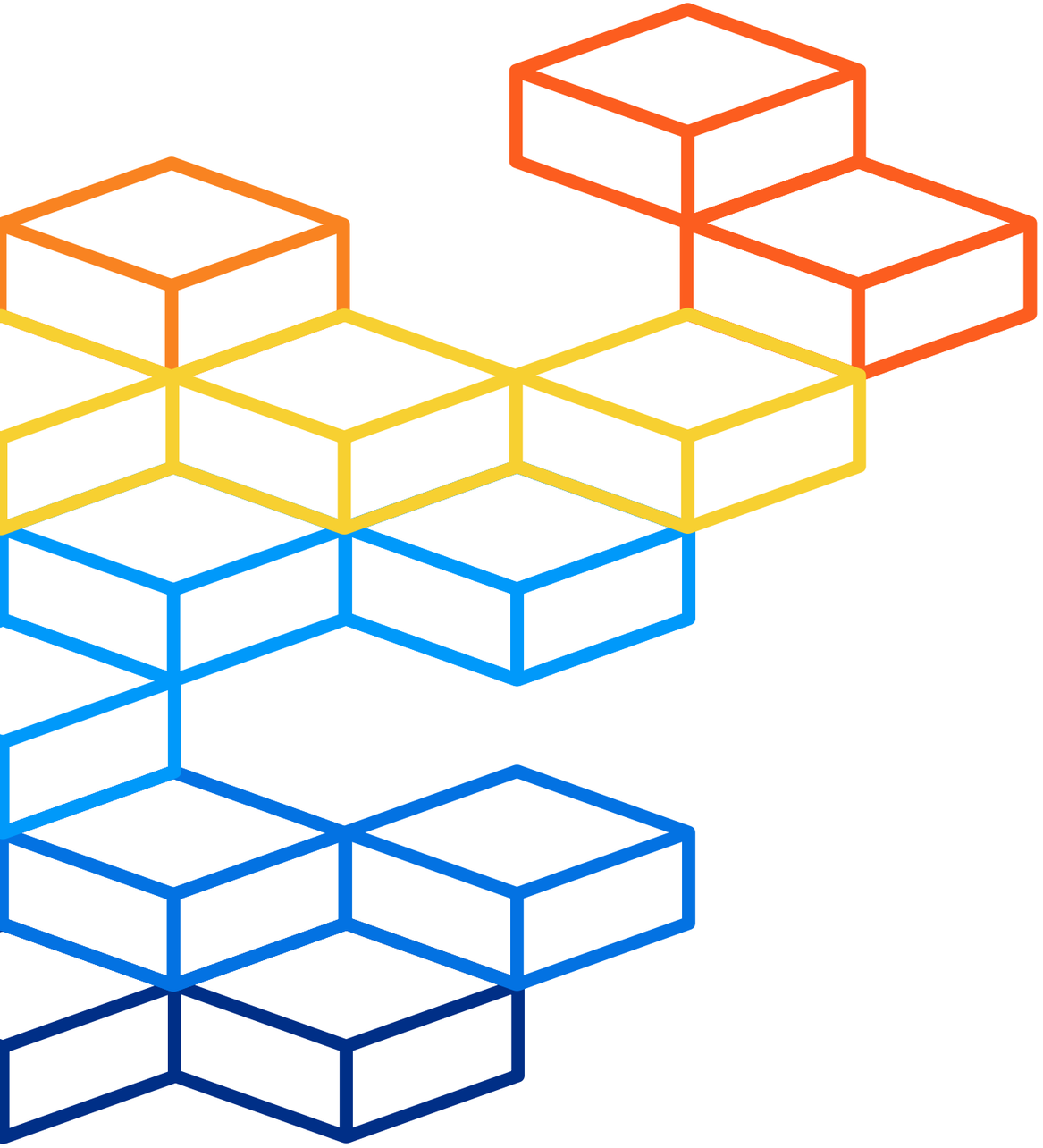
Cold Storage Design — Aerospike Method



1. **Retrieve an ID to return to the edge system**
   - Primary key read
   - Returns all the segments for a user

2. **Write a ID’s segments into the record**
   - Map operations to insert the elements
   - Also do trim by TTL

3. **Find all IDs matching a segment**
   - Scan across all IDs
   - Predicate filters to strip expired segments



Cold storage initial design — systems

- Significant data volume: 50B records x 750 x 20 bytes ~= 750TB raw data
- With RF=2, 50% fragmentation ~= 3PB data storage needed.
- With optimizations, the average record can come down to ~ 11,666 bytes, so 50B is ~583TB with 2.3PB needed

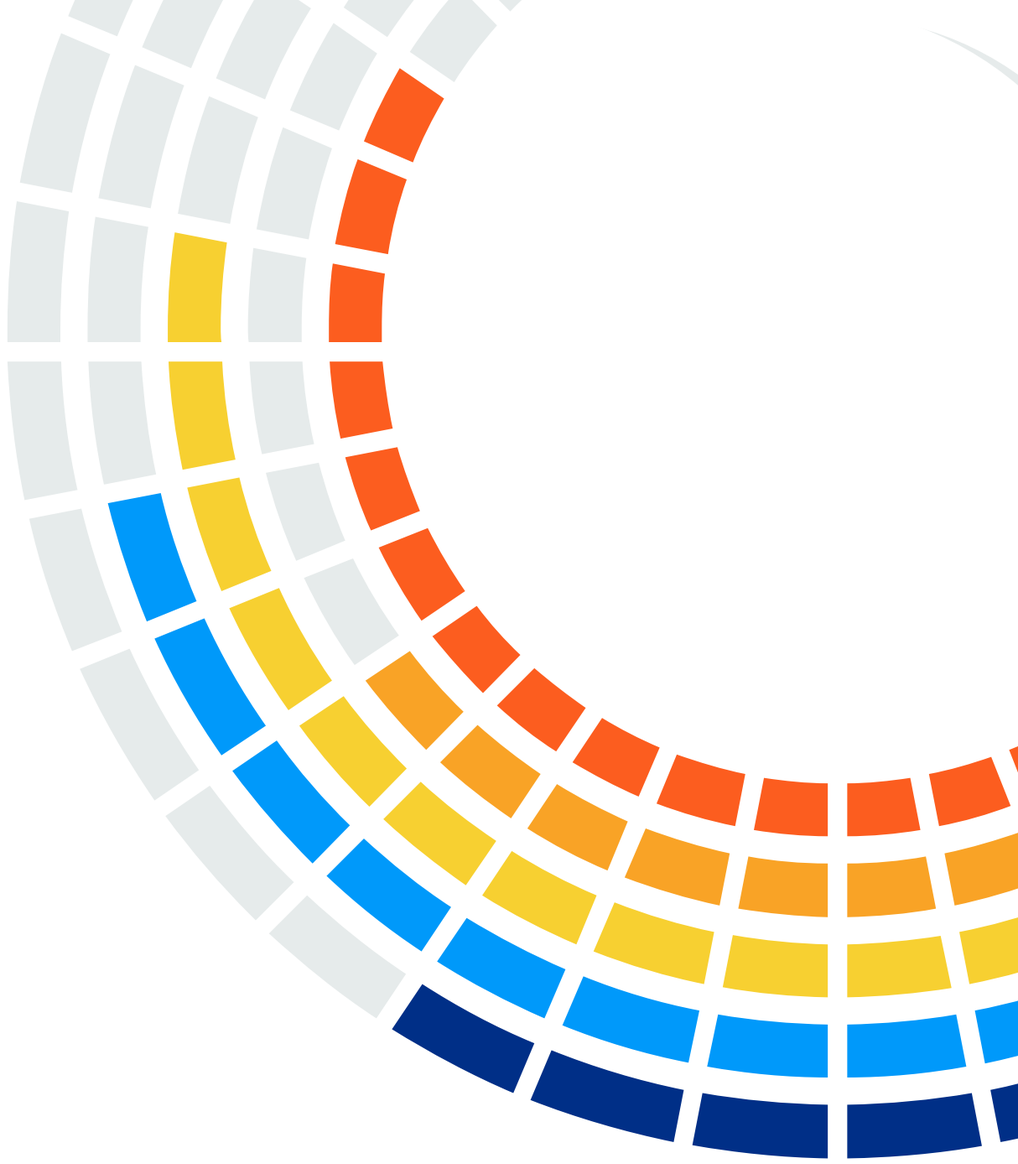
Original AWS sizing:

- 75 md5.metal nodes with compression
- Found we had to increase the cluster to 100 nodes due to IO

theTradeDesk

Cold storage operational data

- 1 MB record size/small update but writing whole 1MB record, IO is rough
- Cold Storage NIC throughput post-compression
  - 750GBPS TX (compressed/ 5x uncompressed)
  - 540GBPS RX (compressed)
- Zstd compressed
- 11,000 cores
- S3 backups - 1pb uncompressed
  - Backup to EBS->S3
  - Great for node recovery/corrupted data
  - Lost 17 nodes, restored and reprocessed in recovery
  - Allows us to do RF=2
- Failure rate high on disk writes - reshuffling data is challenging



theTradeDesk



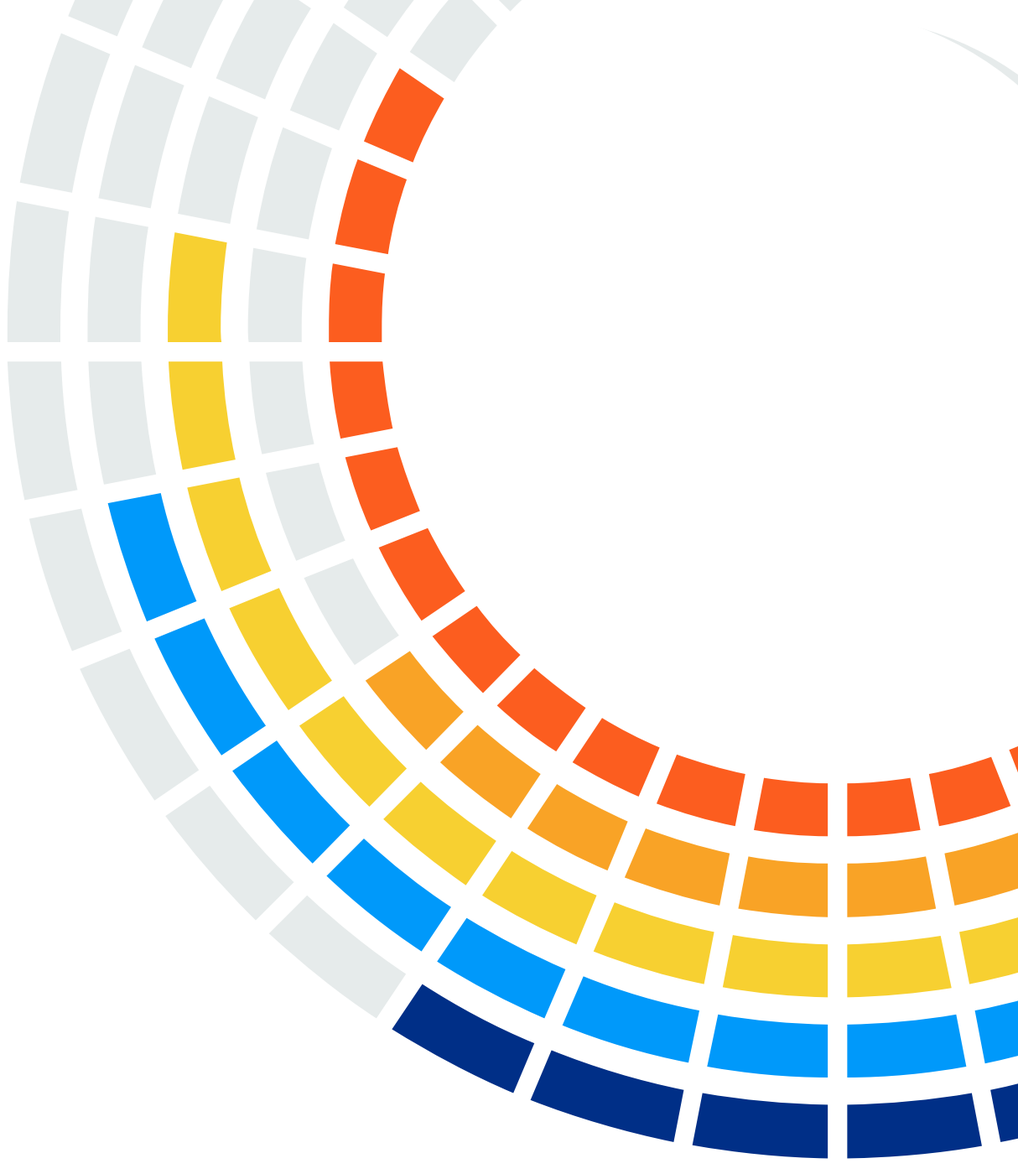
Cold storage operational tips

- Elastic got us started with the full-scale prototype
- Big and gnarly
- ACT - ACT is your friend, run for 24 hours to get confluence of drive/ssd gc
- i3.ens didn’t work well here for the cost – compression requires too much CPU
- Compression required not only for storage; saves us IO/nic
- Watch CPU when using compression
- M5d.metal right combination of CPU/storage/costs
- NVMe drives on m5d.metals get pushed at this throughput
- As low of RF as you can go – crosstalk magnifies scale challenges
- As low of a block size as you can go for hybrid storage models
  – writing 1MB blocks for little updates painful

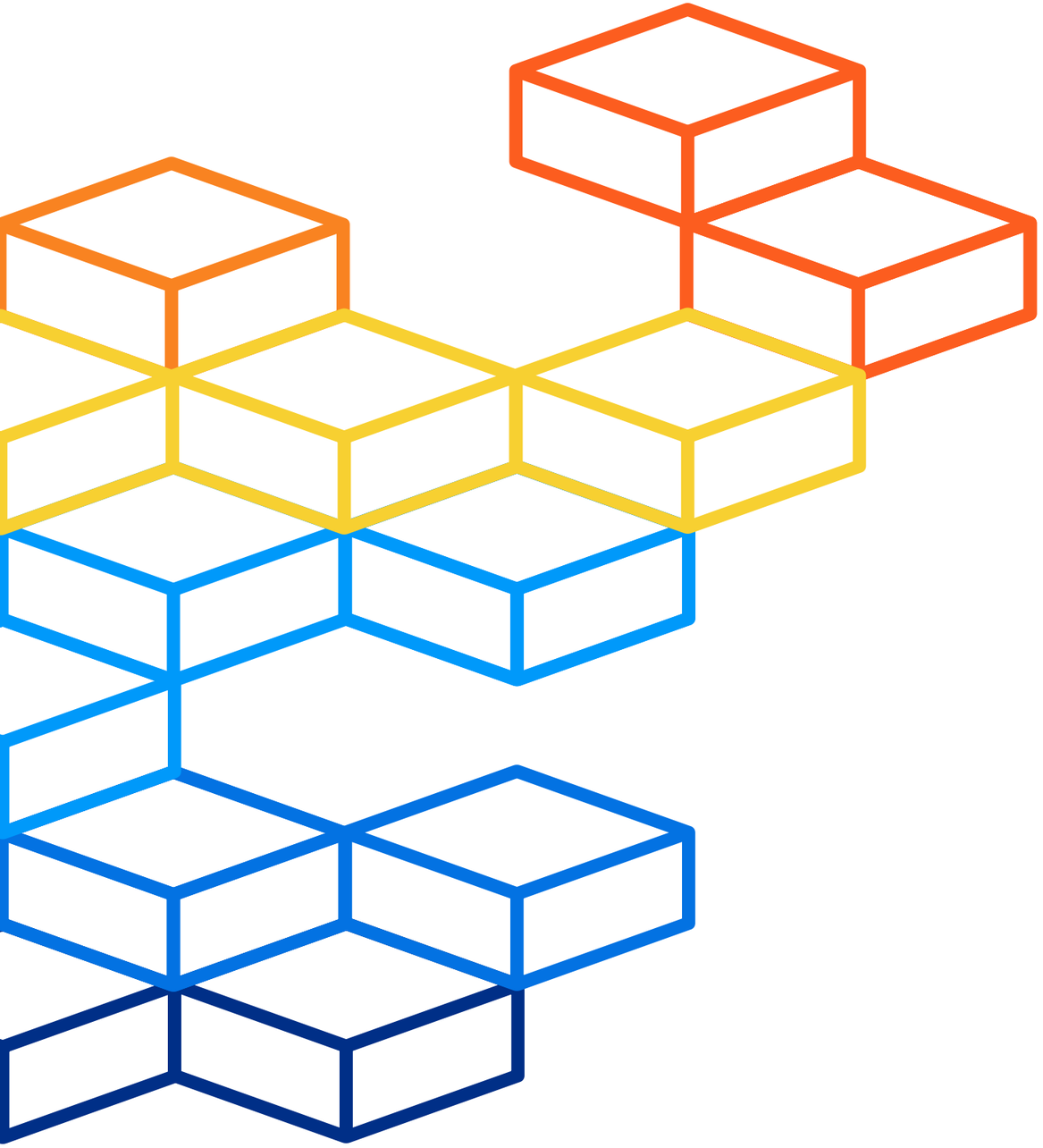
theTradeDesk

Results

- Hot caches contain only the data we need for current ad campaigns
- Hot caches globally refreshed in a short time of new campaign activation
- Data delivery impact on bidding site performance eliminated
- Aerospike spark connector combined with cold storage cluster allows for advanced analytics
- Improved accuracy in understanding reach



theTradeDesk



Lessons Learned

- AWS sizing: Aerospike Certification Tool is critical to understand your workload
- Diversity of EC2 instance types are super helpful
- Elasticity gives us the capability to iterate – our first efforts don’t need to be the ultimate solution
- Application design with Aerospike creates opportunities for incredible optimizations

theTradeDesk

Thank you!

Matt Cochran

Director, Engineering
The Trade Desk

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.