
AWS
re:Invent

ADM201

Contextual targeting and ad tech migration best practices

Tony Lucia
Chief Technology Officer
Integral Ad Science

Danny Rathjens
Senior Director, Technical Operations
Integral Ad Science

AWS re:Invent

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

Agenda

1. Introduction to IAS
2. Our journey
3. What we learned along the way
4. Processing events at scale
5. Contextual analysis at scale
6. Summary

In order for digital media to thrive, buyers and sellers need access to the same measure of media quality.

BUYERS



SELLERS

How we help:

IAS's solutions span media buyers and sellers, reducing discrepancies and creating a frictionless ad environment.

WHO WE ARE



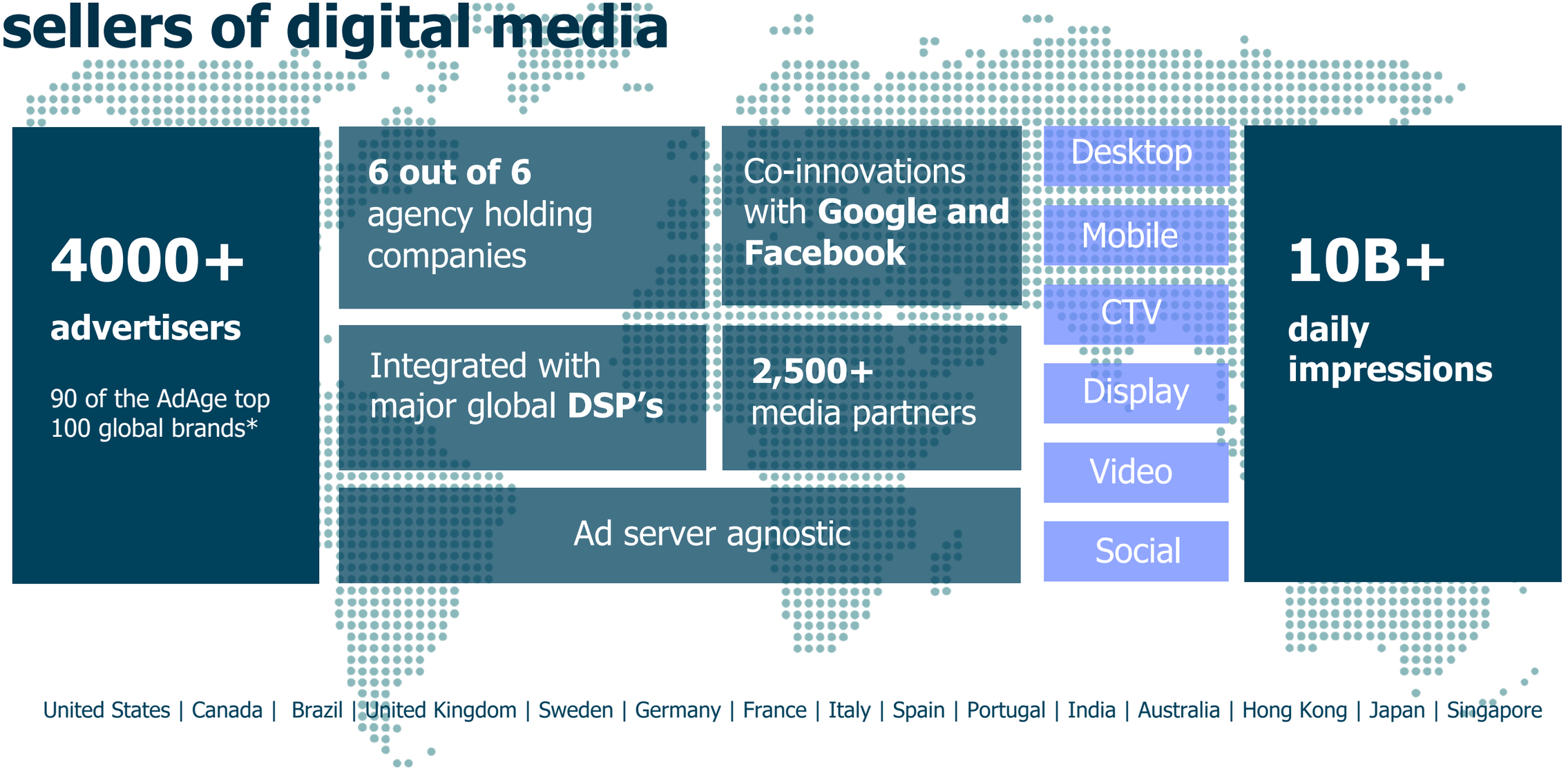
IAS Integral Ad Science

2009	600+	17	12
founding year	employees	offices globally	countries

Integral Ad Science (IAS) is the global leader in **digital ad verification**, offering technologies that drive high-quality digital advertising media.

IAS equips advertisers and publishers with both the technology to protect their advertising investments from fraud and unsafe environments, as well as the insight to capture consumer attention, and drive business outcomes.

Global coverage spanning buyers and sellers of digital media



United States | Canada | Brazil | United Kingdom | Sweden | Germany | France | Italy | Spain | Portugal | India | Australia | Hong Kong | Japan | Singapore

Science + **scale**

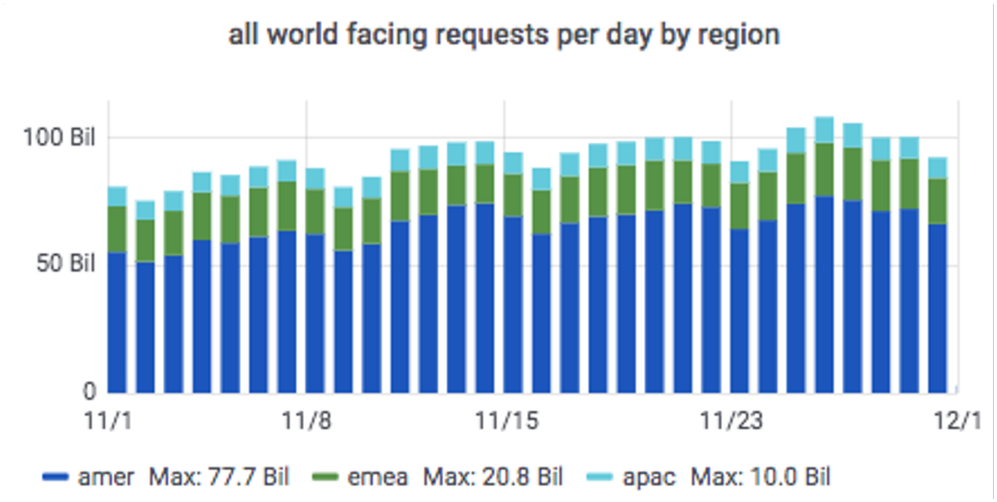
During peak times, our IAS world-facing infrastructure handled over

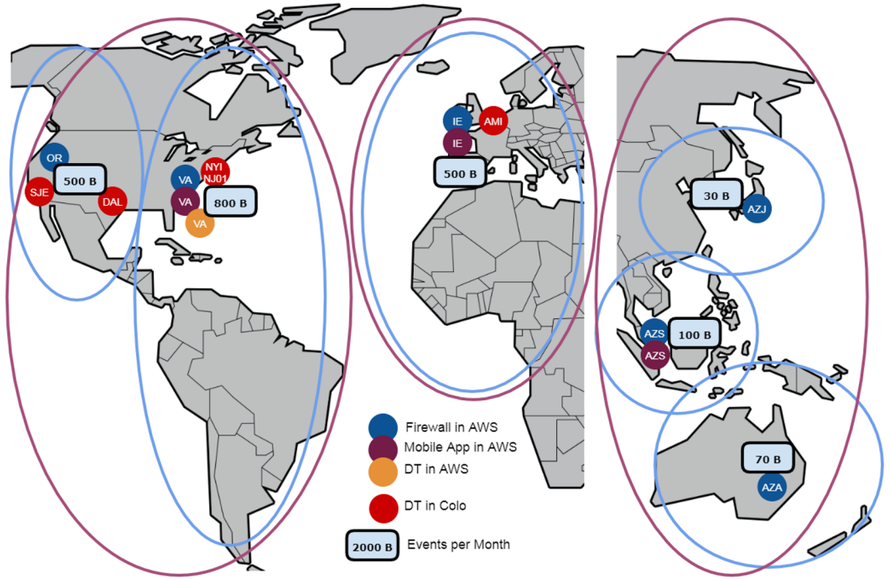
100 billion web transactions per day.

Those transactions produce over

2 trillion events each month

that our internal big data infrastructure processes and analyzes.





What we offer

Verification

Verify your inventory is viewable, fraud-free, and brand-safe

- Ad fraud ✓
- Viewability ✓
- Brand safety ✓

Optimization

Optimize automatically towards advertisers' KPIs

- Yield and efficiency
- Time-in-view
- Brand suitability
- Invalid traffic

Intelligence

Increase revenue potential with contextual intelligence

Smart inventory packaging
Use content segmentation to monetize new channels

Advanced contextual targeting
Command higher CPMs for hyper-targeted ads

Audience engagement
Contextual alignment increases site engagement

Our journey



“Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.”

Steve Jobs
Rolling Stone Magazine, 1994

The journey
IAS environment prior to AWS

- **Data managed**
  - 4 PB average per month
  - 5 PB during peak months
- **Colocation infrastructure**
  - Amsterdam, Netherlands
  - Secaucus, New Jersey
  - Dallas, Texas
  - San Jose, California
- **Capital-intense model**
  - $16M in hardware
  - 3,000+ servers and networking hardware
- **Provisioning / management**
  - 3–6 weeks to procure, rack, and build servers
  - High Mean Time to Repair (MTTR) – No SLA
- **Edge processing (distributed model)**
  - Ever-increasing global footprint built for peak performance
  - 2 trillion+ transactions per month



The journey
IAS challenges / opportunities

- Data-driven business that requires a scalable environment for both high and low-demand business cycles
  - AWS dynamic provisioning allows us to expand and contract infrastructure as needed
  - Engineering teams are now able to improve their SLAs
  - AWS **Spot** market usage allows us to use more infrastructure as needed at significantly deep discounts
- Capital-intensive business model – leasing hardware for peak times
  - AWS cloud services provider allows us to consume only what we need
  - We improved our capital asset allocation and financial forecasting
  - We improved our visibility into the true cost of ownership for our products
- Provisioning hardware and setup takes too long
  - AWS Infrastructure as Code (IaC) allows us to reduce hardware provisioning from previously **3–6 weeks to several hours**
- Business agility
  - Technology procurement is no longer a blocker for our agile teams



The journey
What we considered

- Understanding our cost drivers
- Capital allocation
- Business continuity
- Customer impact
- Business agility



Why we ultimately decided to go all in

We kept buying a lot of hardware every year for growing peak traffic, which then sat idle when traffic dipped

- No **Spot market** available when you run own your data center
- AWS On-Demand services could help solve our capital asset allocation issue

It takes us several weeks to buy and provision new hardware for new products

- This is no way to support agile teams
- AWS could help us improve our time to market



Business agility

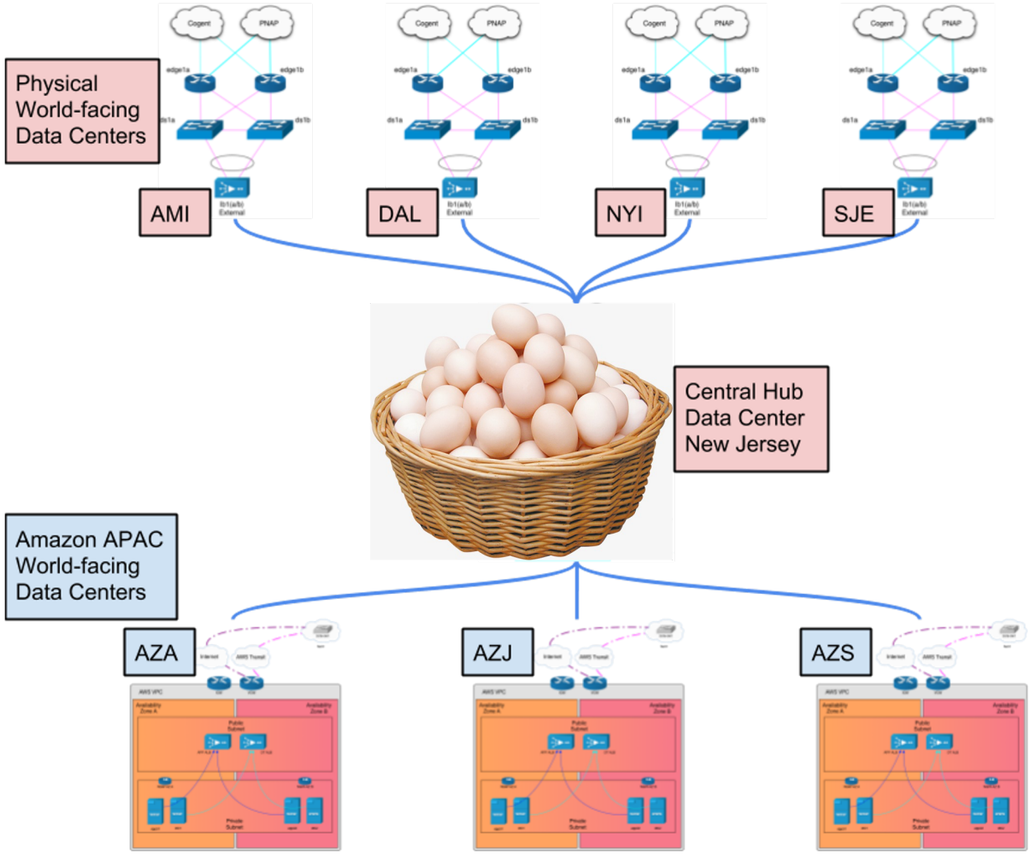
We saw an opportunity to change the game

We use a hub-and-spoke model

“Our eggs in one basket”

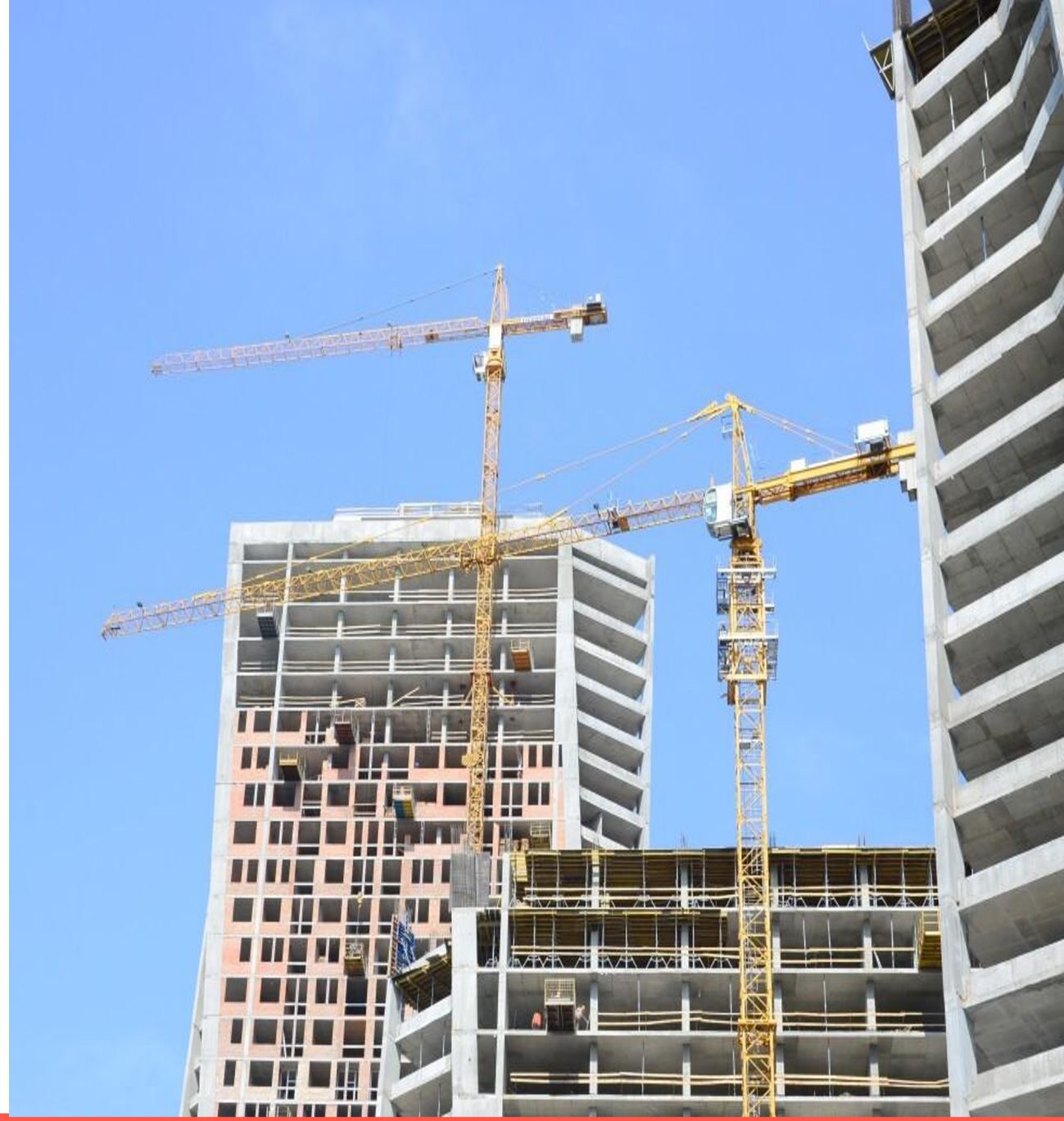
These eggs are a huge pile of hardware:
servers, networks, wires…

Growth requires IAS to buy more and more hardware every year, slowing down how quickly we can expand our products and services



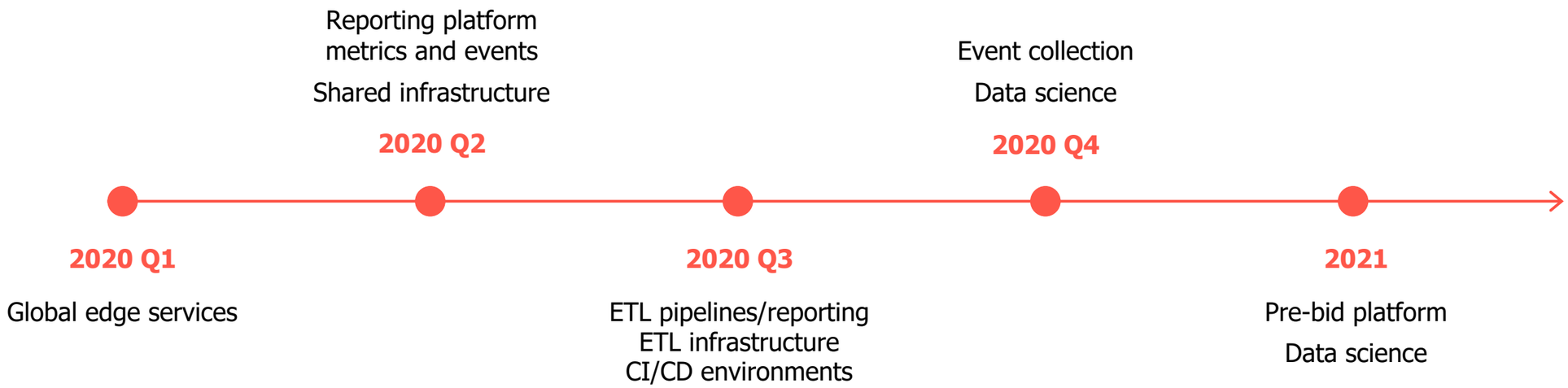
Bootstrapping our migration

1. Gained buy-in, including **financial justification** and approval from business leaders
2. Communicated the strategy to the entire organization
3. Identified AWS partner to perform a **migration readiness assessment** (independent view)
4. Implemented a company-wide **AWS training program** to educate staff
5. Partnered with AWS to provide guidance throughout the migration (solution architects)
6. Worked with AWS to create **immersion days**
7. Leveraged AWS professional services to build scaffolding for new services (e.g. data lake)
8. Drove a "lift-and-shift" approach to accelerate the migration and / or build new
9. **Optimized** post-migration
10. Embraced **AWS innovation process**

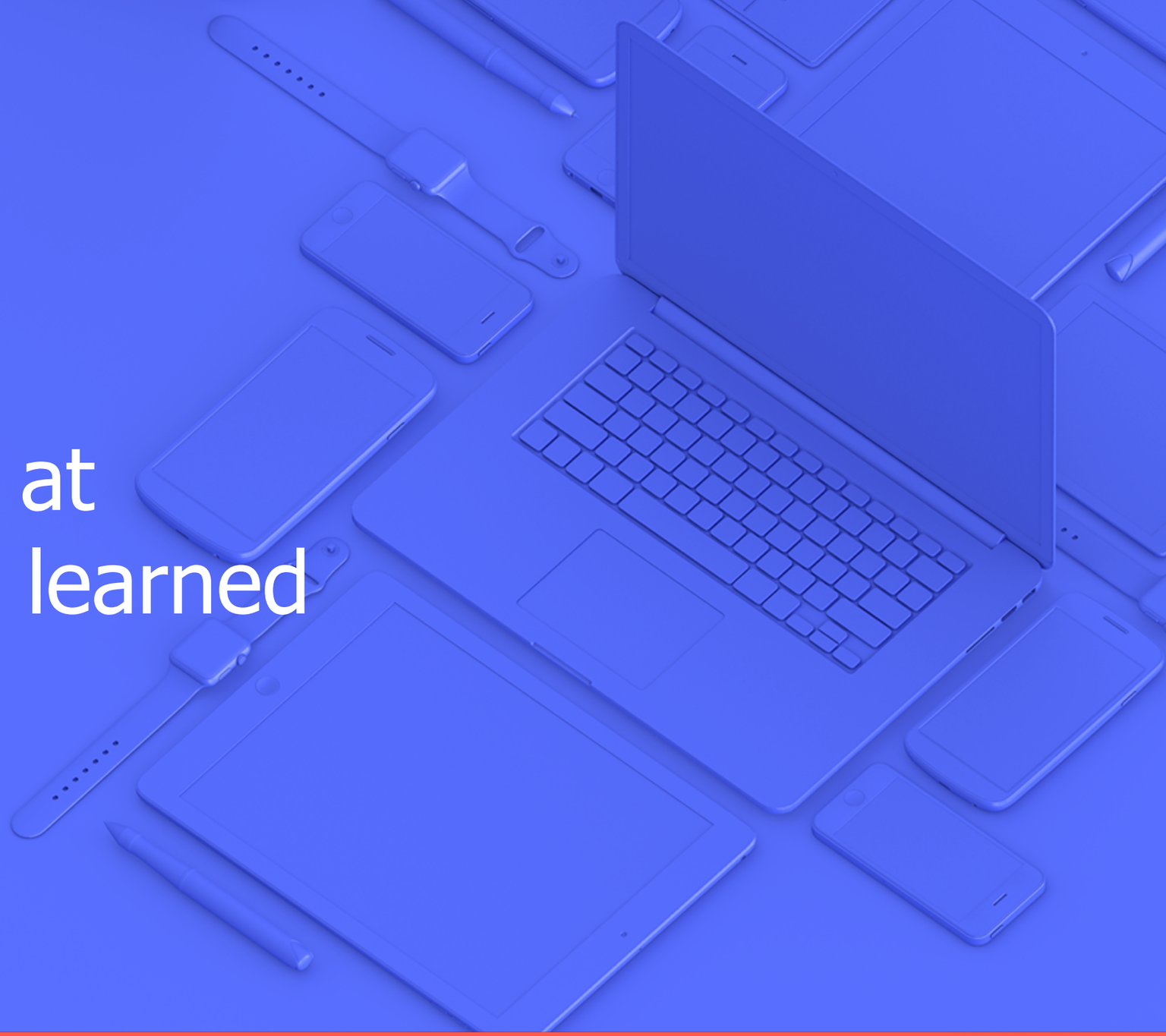


Migration journey

On track to complete the bulk of the migration in 2020

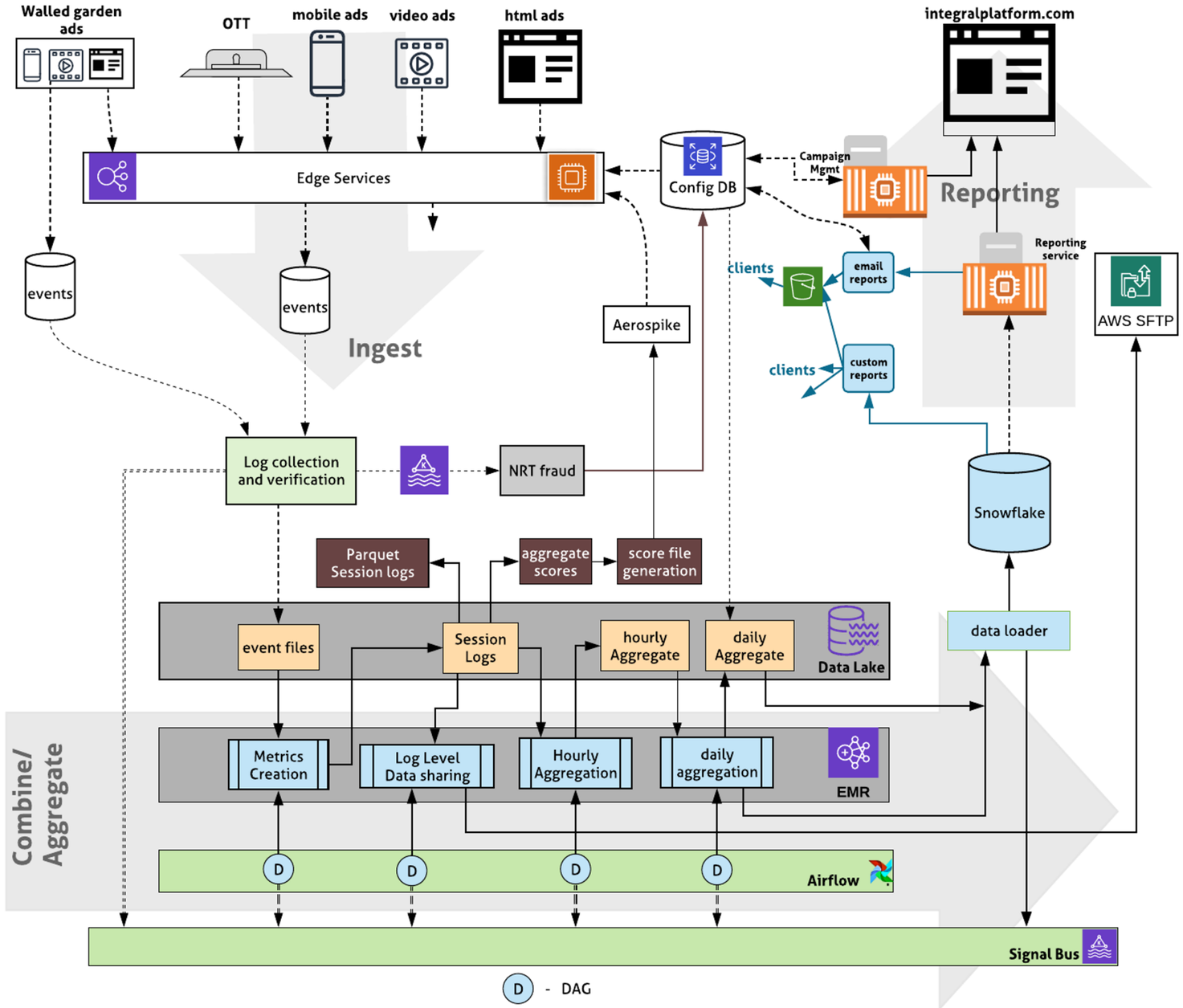


Processing events at scale, and lessons learned during migration



Workflow

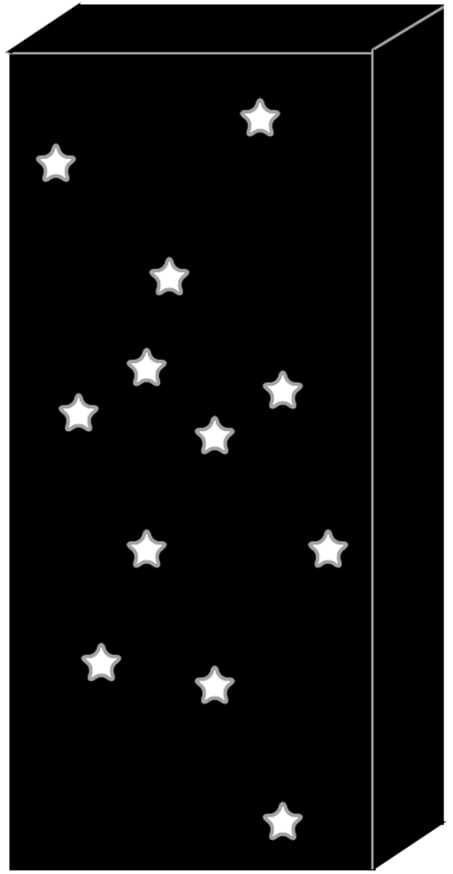
- **Serve** answers to brand safety and code to measure and verify ads
- **Ingest** data from several device types and formats and partners
- **Combine and aggregate** event logs for specific impressions and campaigns
- **Generate** scores to update our applications
- **Report** the results to our clients



Moving away from the monolith

Our primary world-facing application servers in the collocation handled:

- SSL termination
- Static content
- API calls from direct clients
- Brand safety and verification requests from millions of web browsers
- Access log compliance archival



Moving away from the monolith

We were able to optimize our architecture by splitting the monolith



Elastic Load Balancing

SSL termination is now handled by AWS ALB

Improving elasticity



Amazon CloudFront

Static content is now served via CloudFront

Improving latency and elasticity



Amazon EC2

Separate dedicated pools of EC2 instances to handle our API calls separately from our brand safety and verification requests

Allowing for keepalive optimizations and scaling



Amazon Simple Storage Service (S3)

Access logs disabled on servers and ALB S3 access logs stored in S3 Glacier Deep Archive for compliance

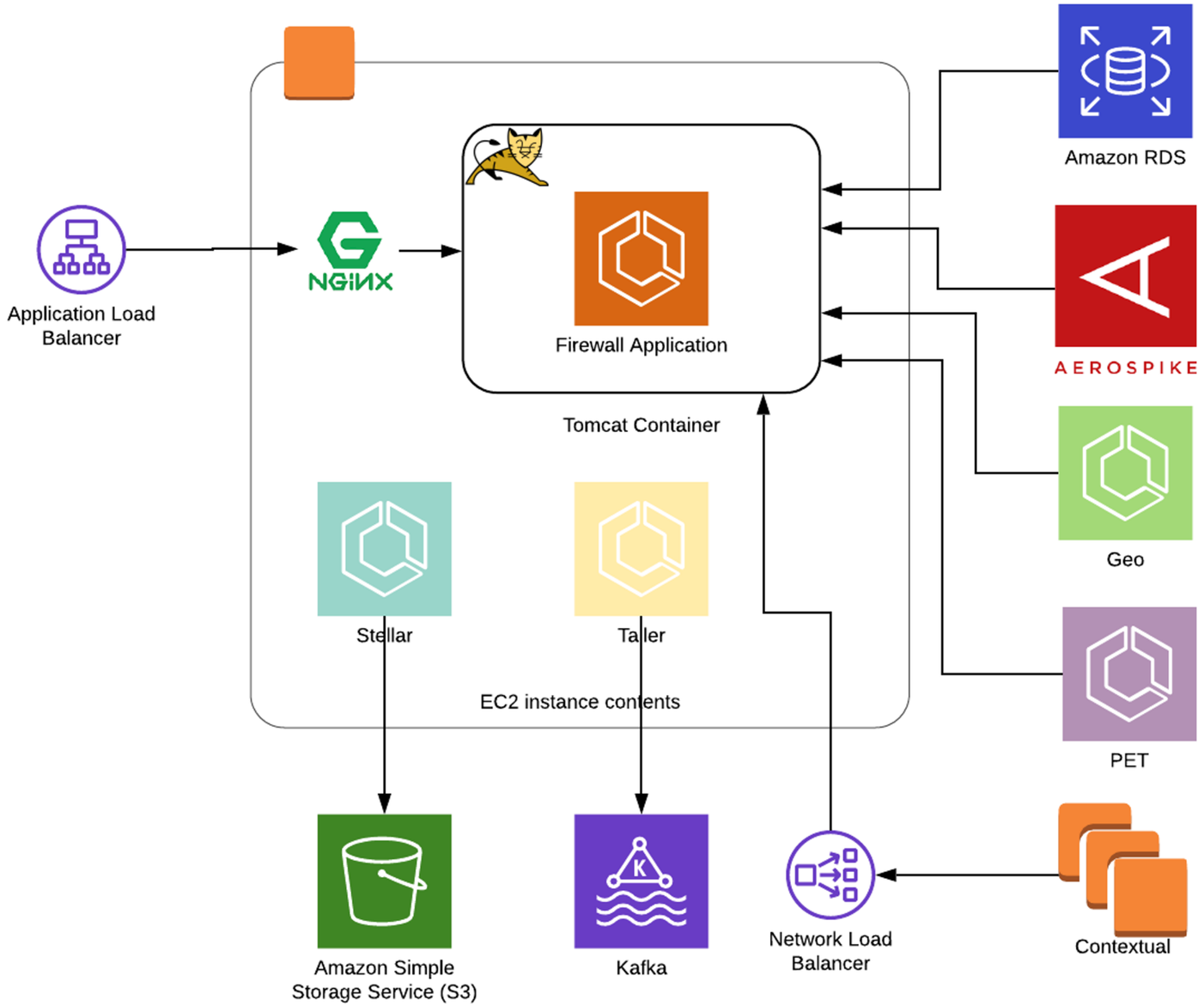
Lowering instance storage and improving performance

Edge architecture

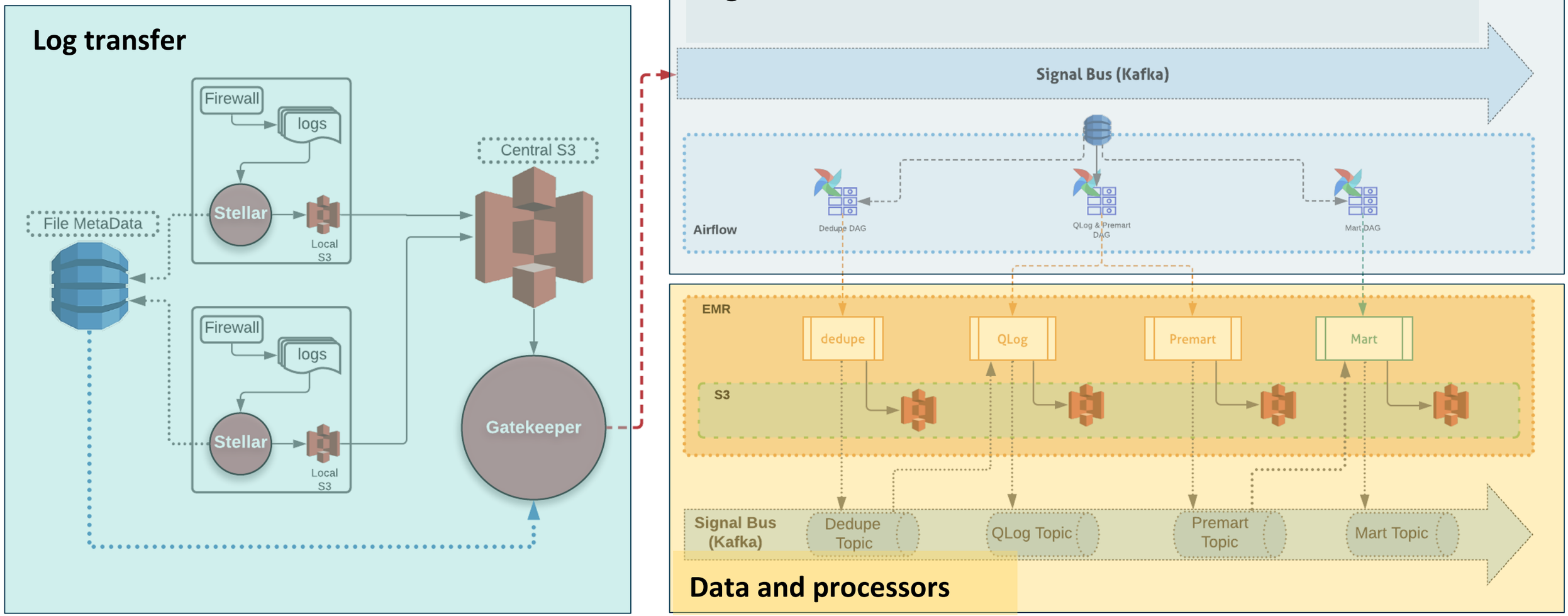
Nginx
Tomcat
Java application

RDS MySQL
Aerospike
Geolocation

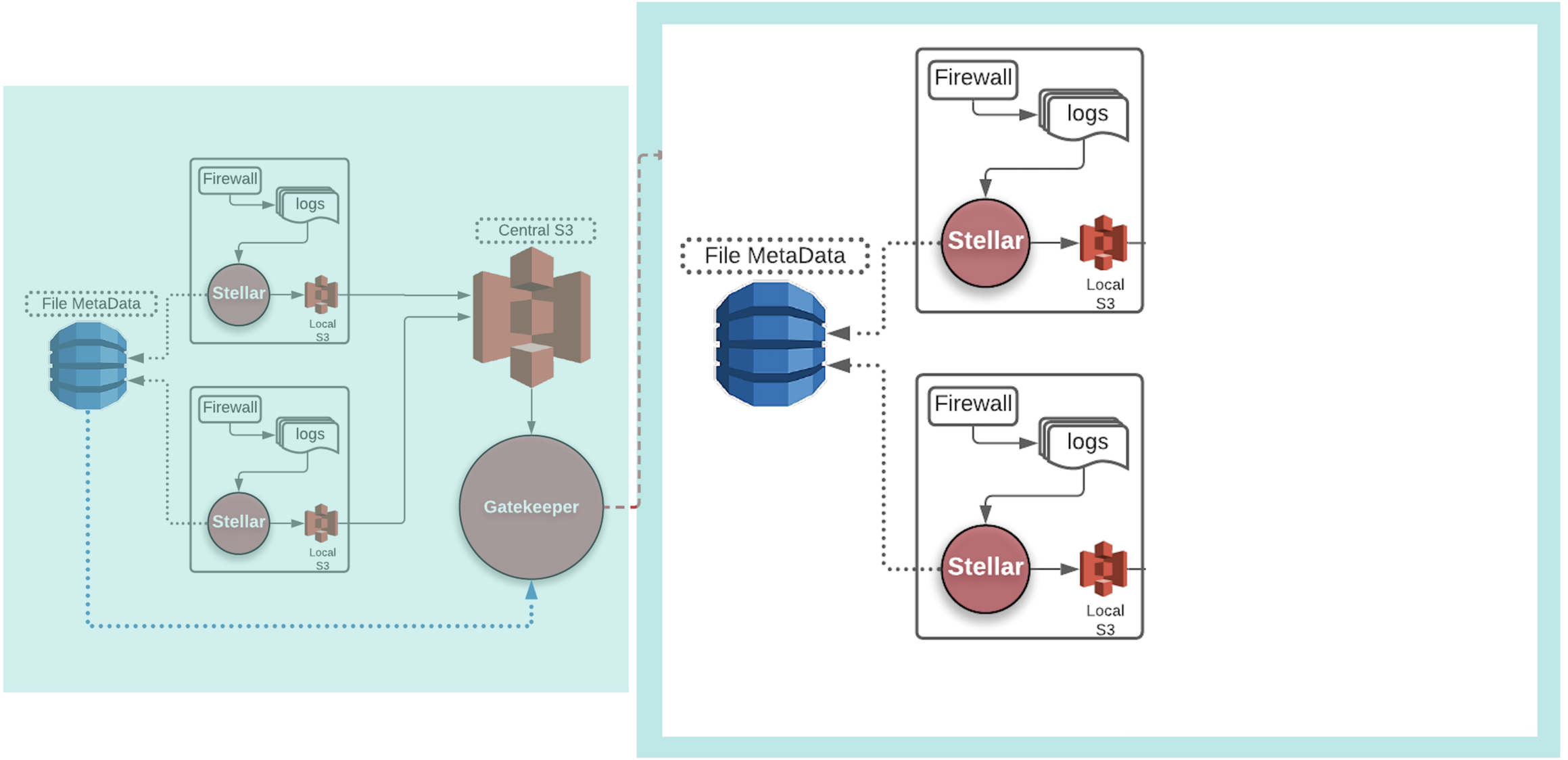
Contextual



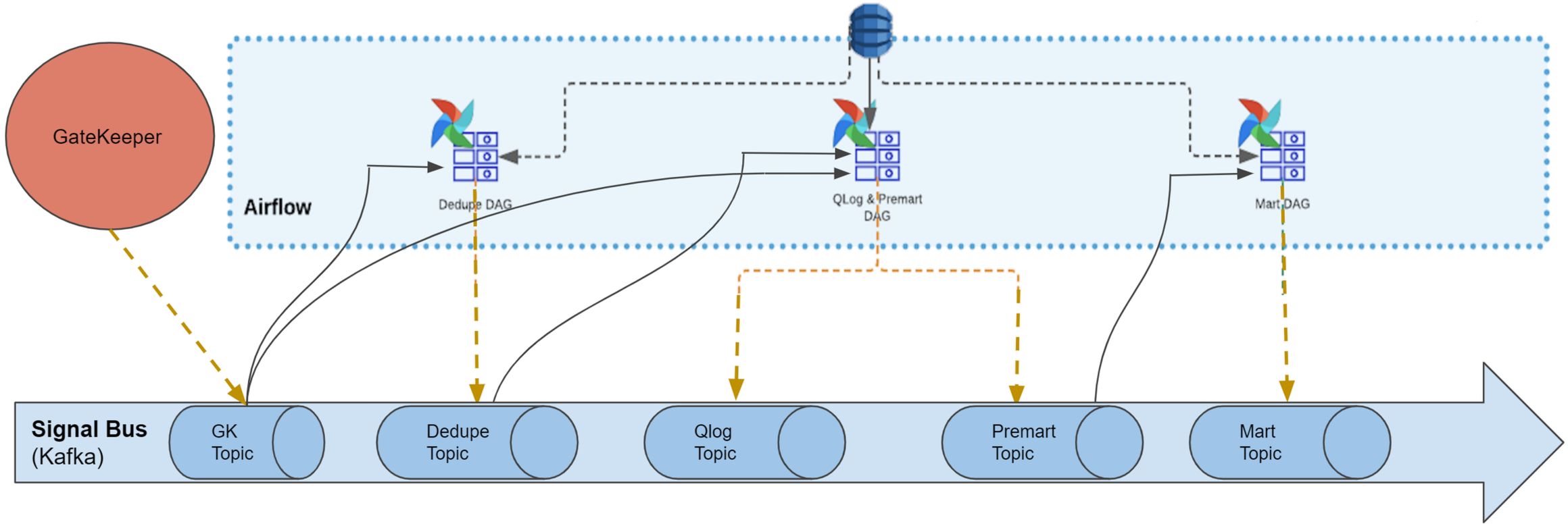
Architecture



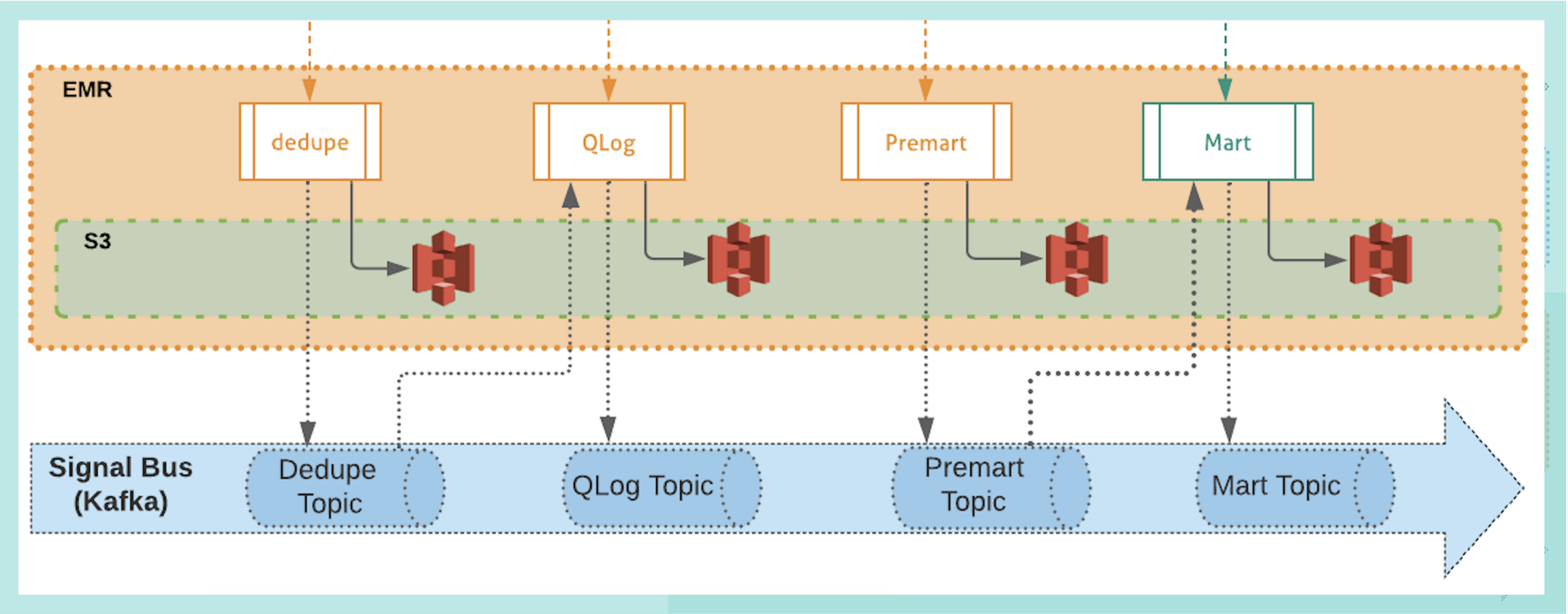
Log transfer



Signal bus: How does it work?



Data and processors



Issues we encountered: Amazon S3 eventual consistency intricacies

"A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, **Amazon S3 might return the previous data."**

```
PUT foo/bar.txt Content1
PUT foo/bar.txt Content2
GET foo/bar.txt                 # returns Content1, Content2
```

"A process writes a new object to Amazon S3 and immediately lists keys within its bucket. Until the change is fully propagated, **the object might not appear in the list."**

```
PUT  foo/bar.txt
PUT  foo/baz.txt
LIST foo/         # returns [], [bar.txt], [baz.txt] or [bar.txt, baz.txt]
```

Amazon S3 eventual consistency intricacies

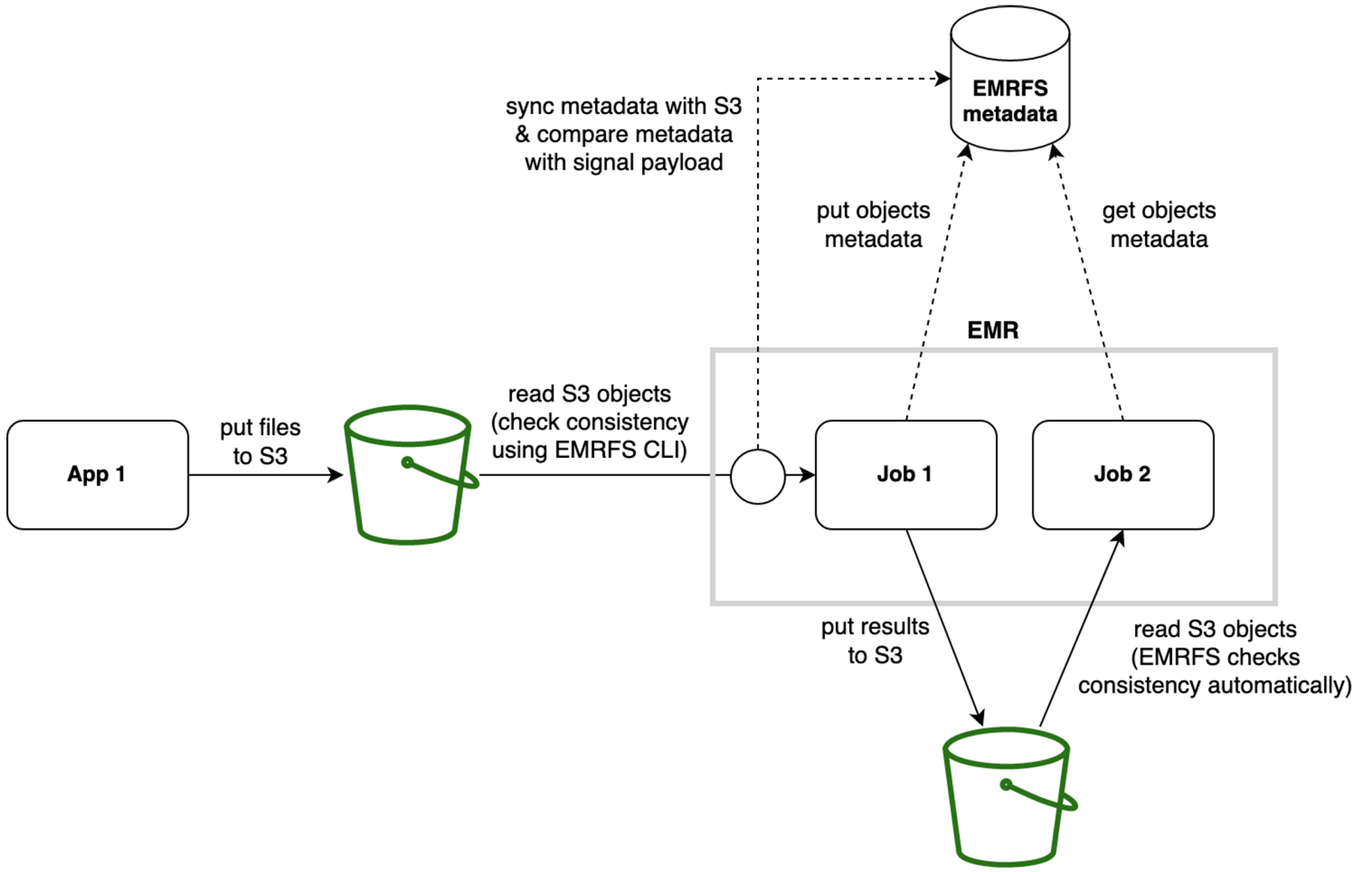
Restatements require reprocessing the same time buckets—how do we deal with this?

- Data must be immutable
- "Directories" must not be overwritten
- Data location is no longer static—each job run must write into a unique location
- Signal payload contains data location

```
{
  "dataReadyFrom": 1600934400000,
  "dataReadyTo": 1600938000000,
  "dataLocation": {
       "DEDUPE": {
       "path": "s3://iastf-pipeline-intermediate-ue1-de-prod/dedupe/date=20200924/
hour=04/RES-73-1600940595390",
       "fileCount": 1001,
       "totalFileSize": 229907967471
       }
  },
  "createdAt": 1600942241282
}
```

Unique run ID

How to ensure data consistency when Amazon S3 LIST may not be?



EMRFS provides a way to ensure consistency of data between EMR jobs by using Amazon DynamoDB for the metadata

EMR self-provisioned DynamoDB problematic

Worked around with separate DynamoDB

What about data written from outside the EMR cluster?

EMR: Hiccups faced during AWS migration

Resource limits on

- Amazon EC2, IP addresses, Route 53 DNS entries, and even Amazon EBS volumes

Capacity constraints

- EMR limited to a single AZ, and using instance groups made us decide which AZ without knowing which had more capacity
- Instance groups limited us to specific instance types

AWS improvements that helped us

- Instance fleets allowed us to let AWS choose the right AZ and use up to 15 instance types
- Falling back to On-Demand if Spot is not provisioned quickly – more capacity pools
- Understanding details of how AWS reserves capacity – more for smaller instance types
- Managed scaling

The magic of EMR-managed scaling



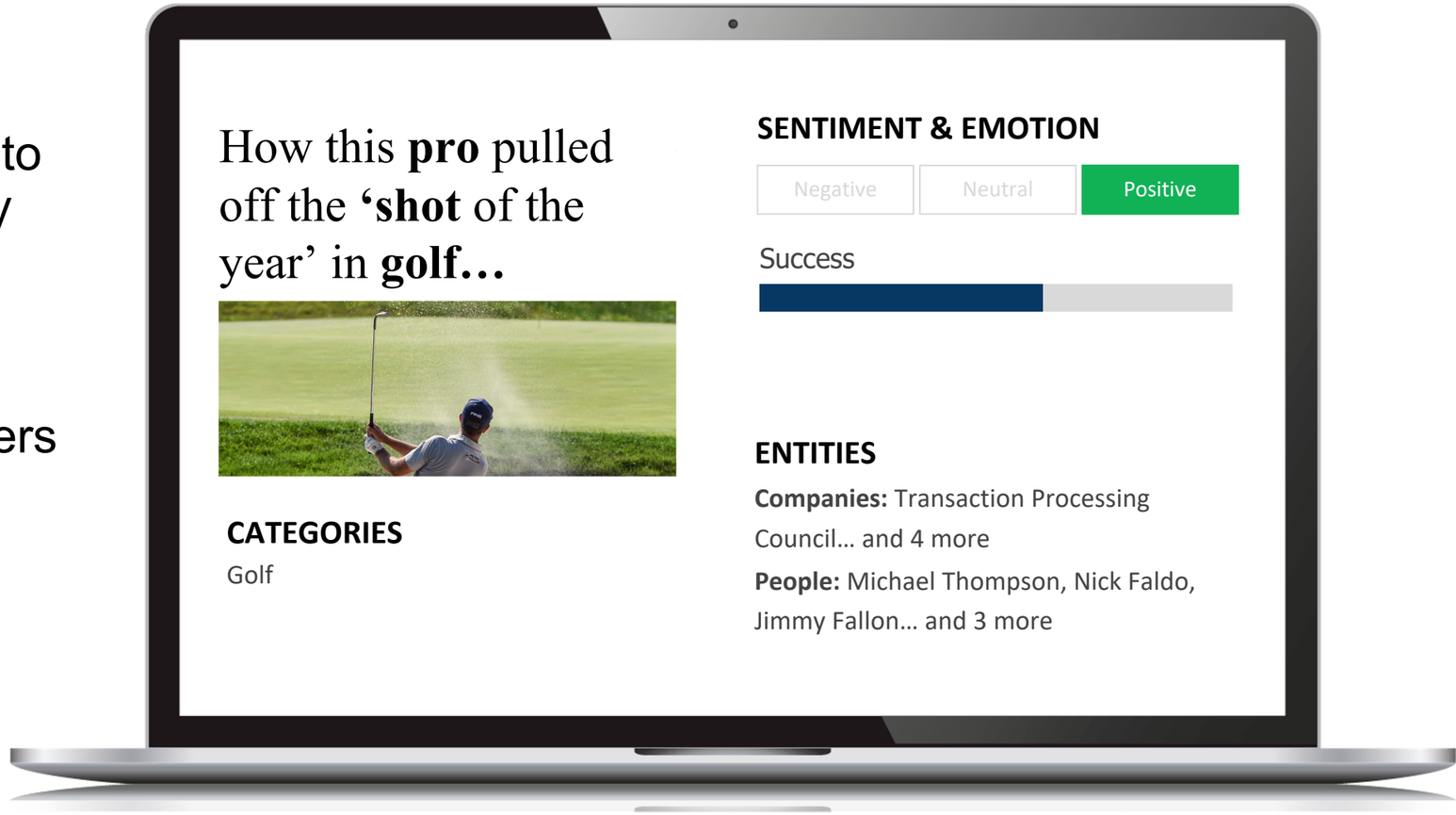
Running contextual analysis at scale



Contextual business opportunity

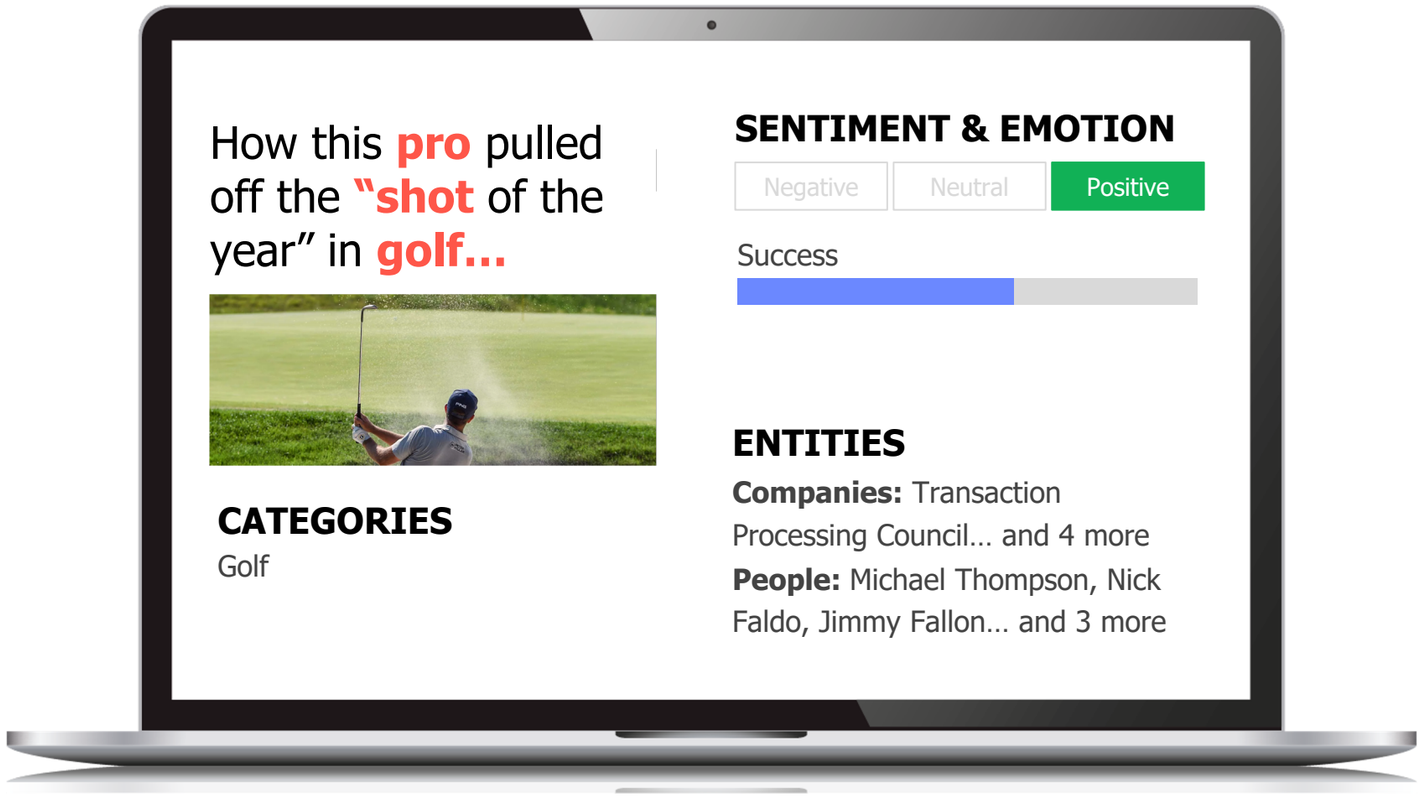
Transforming an acquisition into an immediate benefit for IAS customers

- **Opportunity**
  - Solution that classifies web pages into topical categories with high accuracy
  - 32 languages
- **Challenge**
  - Monolith technology based on manually provisioned, on-prem servers
  - Not easily scalable, not global
- **Solution**
  - Fast roll-out using AWS global edge capabilities
  - Environment optimization in phases to improve the cost model
- **Status**
  - Fully implemented in 3 months



IAS contextual solution

Precise classification of content at scale is the key factor for enabling control



- **100+ industry vertical, seasonal*, and topical** segments available today for targeting and avoidance
- **Brand-specific** content avoidance
- Semantic technology powered by the **largest knowledge graph** in ad verification
- **Sentiment and emotion** detection
- IAS is **"42% more accurate** than the next best provider" at classifying online content**

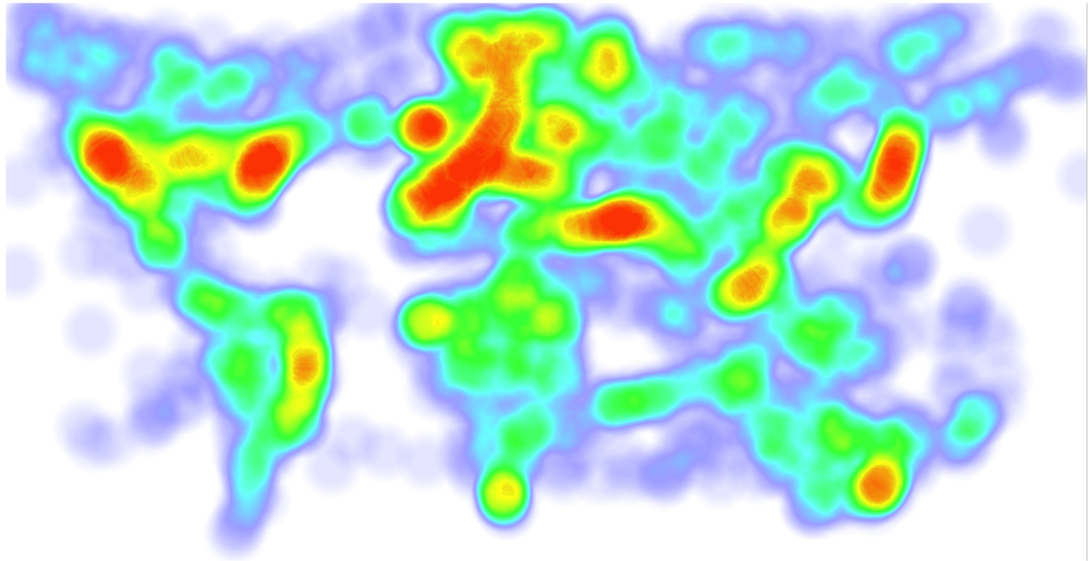
  - THE OZONE PROJECT

IAS differentiator

*Seasonal segments only available for positive targeting

**According to The Ozone Project

For every web page/content item…



- Morphological analysis
- Syntactic parsing
- Semantic analysis
- Categorization

1) Deep linguistic analysis relying on large knowledge graph for a few languages
2) Machine translation for other languages

Advanced technology that understands...

Word forms

Dog, dogs, and dog-catcher are closely related

Parts of speech

“There are over forty rows in the table” — rows = noun vs.
“She rows five times a week” — rows = verb

Relationships

“John Stiles, represented by attorney Jeff Doe, is married to Martha Rivera”

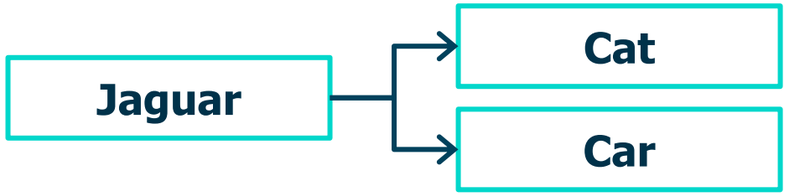
Martha is married to John, not Jeff

Context of key words

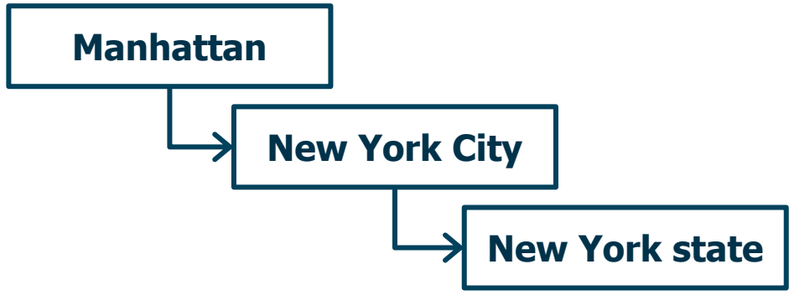
"I used beef stock for my soup” vs. “The company keeps a lot of stock on hand”

Context matters

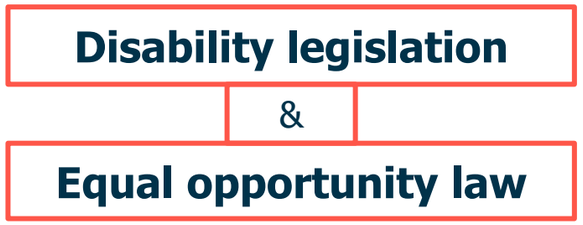
Same words, multiple meanings



Different words, related meanings



Different words, same meaning



Incomplete phrase



Sentiment analysis examples

2019 was a **good** year for us. This year so far has not been so **good,** and we **fear** it's going to get **worse.**

Negative | Neutral | Positive

2019 was a **bad** year for us. This year so far has been **great,** and we hope it's going to **get better.**

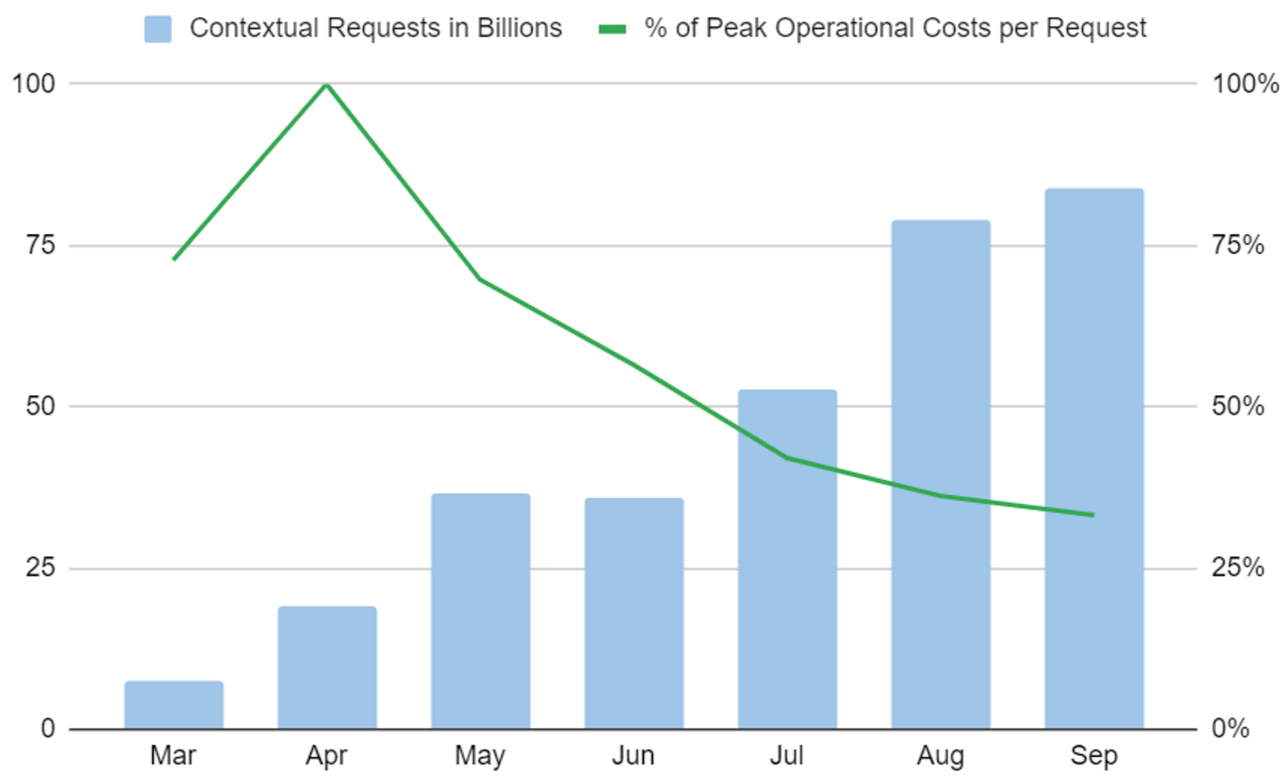
Negative | Neutral | **Positive**

2019 was a bad year for us. This year so far has been great.

Negative | **Neutral** | Positive

Contextual integration operational costs

- **Move fast** strategy
- **Get it working,** then optimize
- **Initial scaling** increased operational CPM highlighting inefficiencies
- **Architectural changes** and savings plan optimized resource usage and costs
- **Operational CPM** was more than halved



Contextual architecture objectives

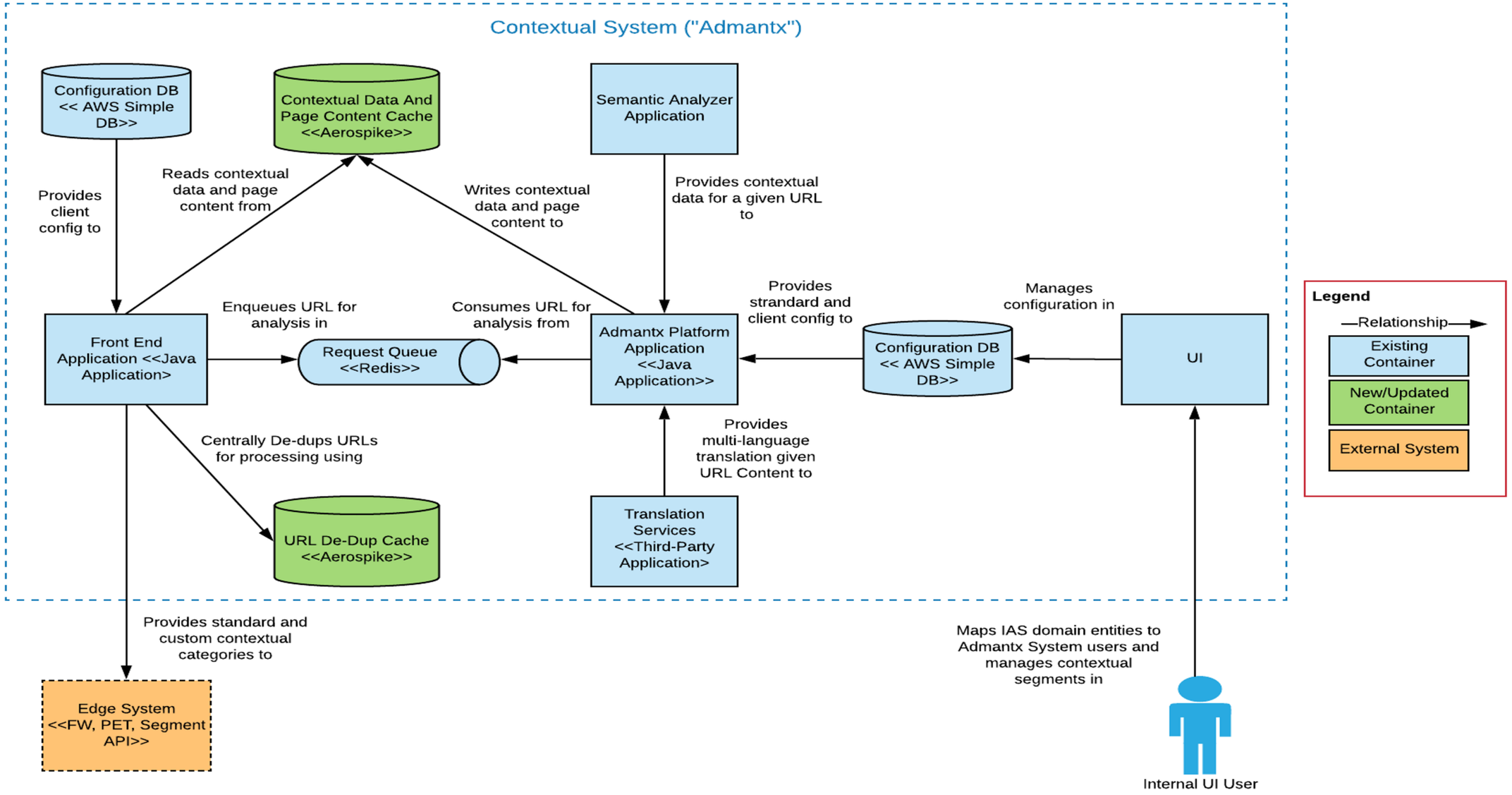
Integrate contextual system into our existing platform and scale it up dramatically

Provide known contextual scores in under 10ms for 99% of requests

If we don’t have scores yet:

- Fetch the content from the website
- Translate it if we don't have a native language model
- Apply contextual (morphological, syntactic, semantic, categorization) analysis
- Write the scores to cache in under 10s

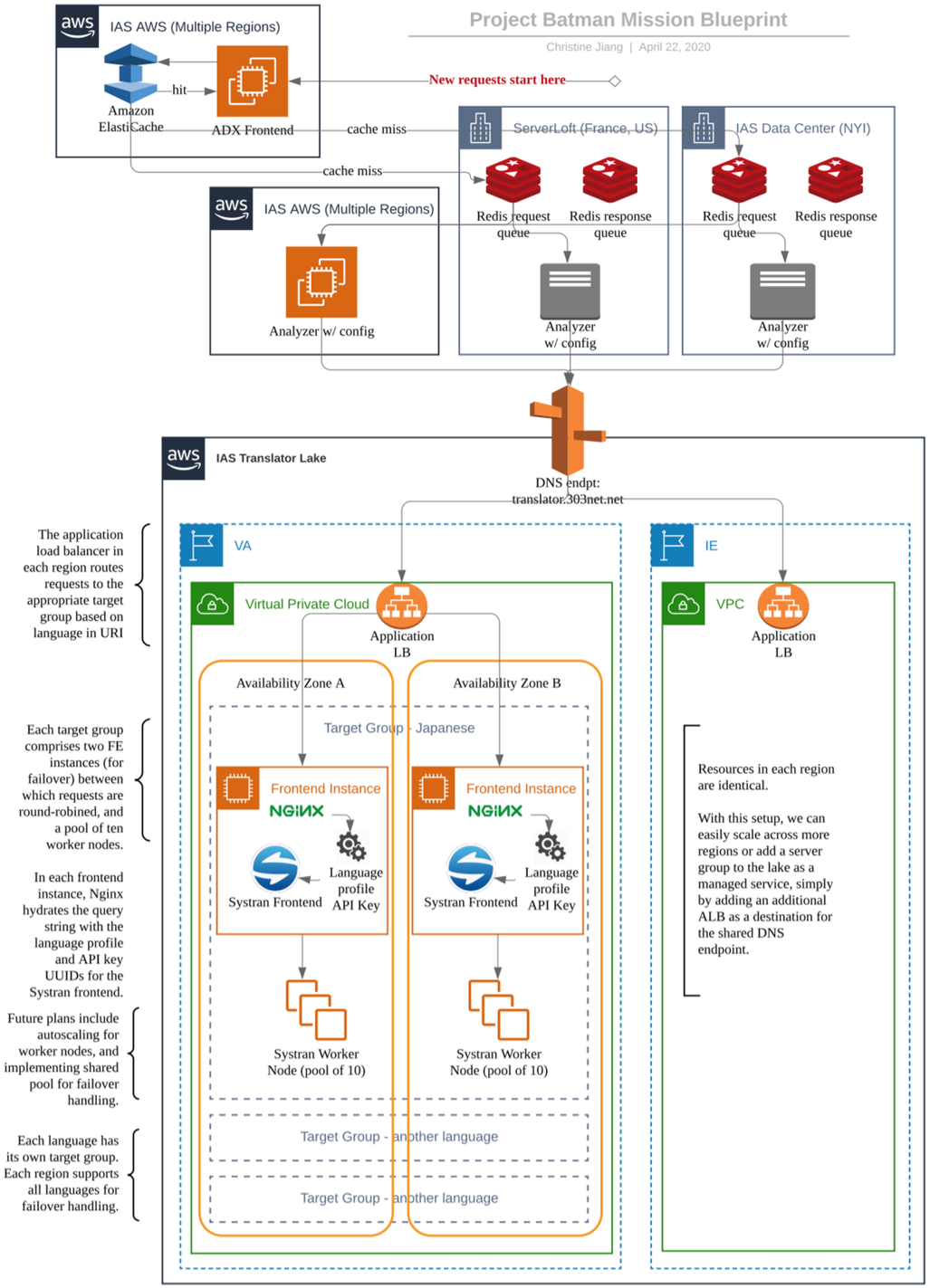
Contextual architecture



Hybrid architecture

A decoupled frontend and backend architecture allowed us to partially migrate and continue to leverage existing hardware while expanding and optimizing our components in AWS

The translation lake was a great example of this cloud optimization



Aerospike

By treating an Availability Zone as a rack, we can leverage the Aerospike Rack Awareness feature to direct calls from our application servers to the Aerospike node within the same AZ.

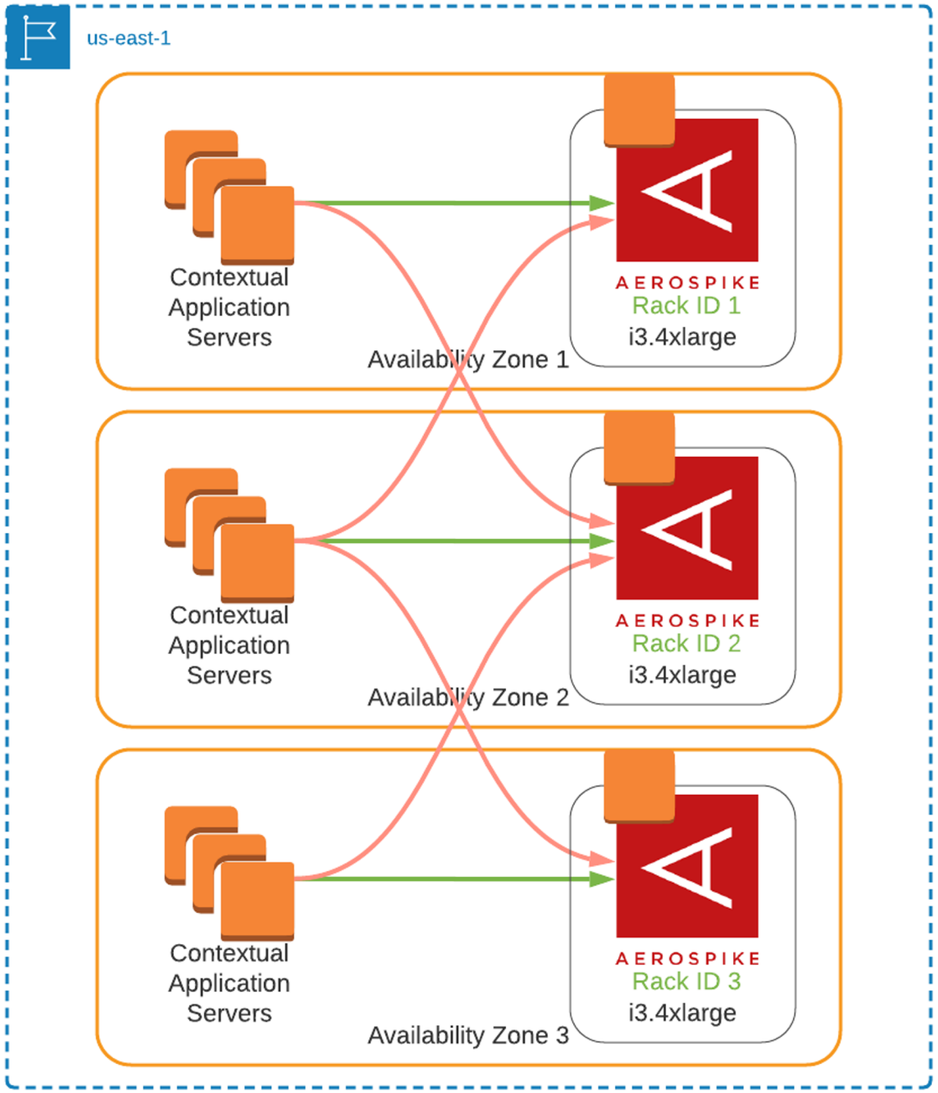
Latency improvement

Cost improvement by avoiding cross-AZ network charges

Replication factor of 3 trade-off

-

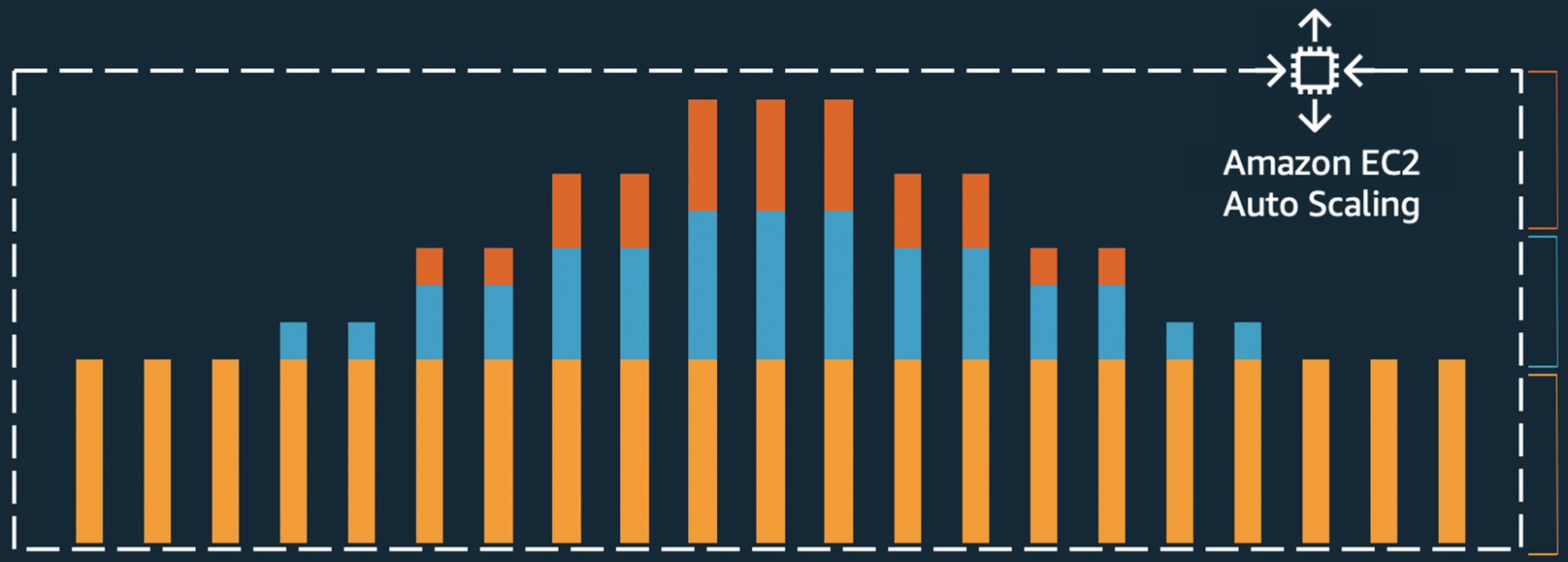
XDR - cross region replication



Keeping costs under control



Optimal reservation ratios



Scale using Spot for short-lived, fault-tolerant, stateless workloads

Scale using On-Demand for short-lived, stateful workloads

Use RIs and Savings Plans for known/steady-state workloads

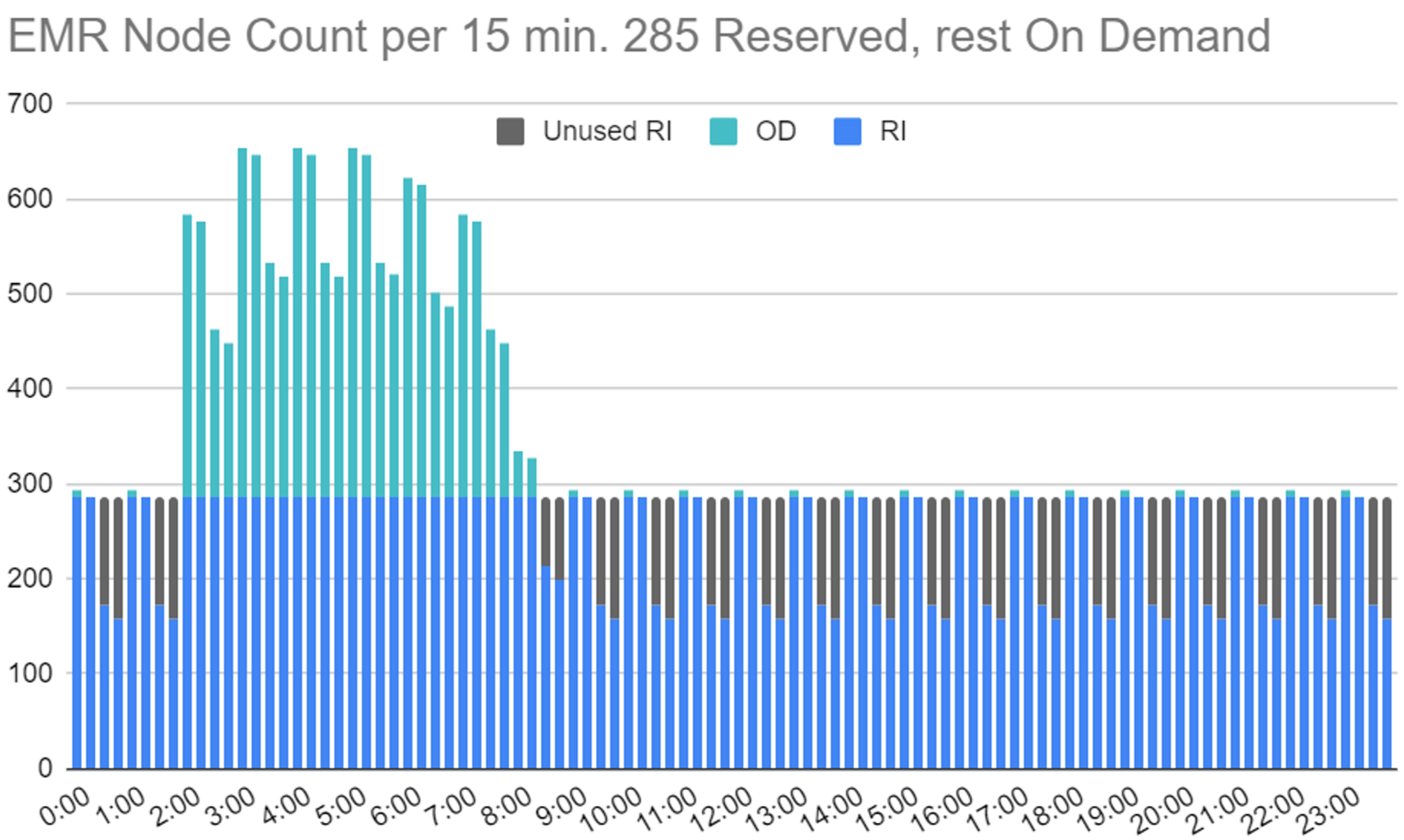
AWS SERVICES MAKE THIS EASY AND EFFICIENT

- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Service (Amazon EKS)
- AWS Thinkbox
- Amazon EMR
- AWS CloudFormation
- AWS Batch
- Amazon SageMaker

Optimal reservation ratios

Committing to usage at the optimal ratio with some unused RI/SP daily gave us an additional 10% savings versus reserving capacity for only the lowest daily usage

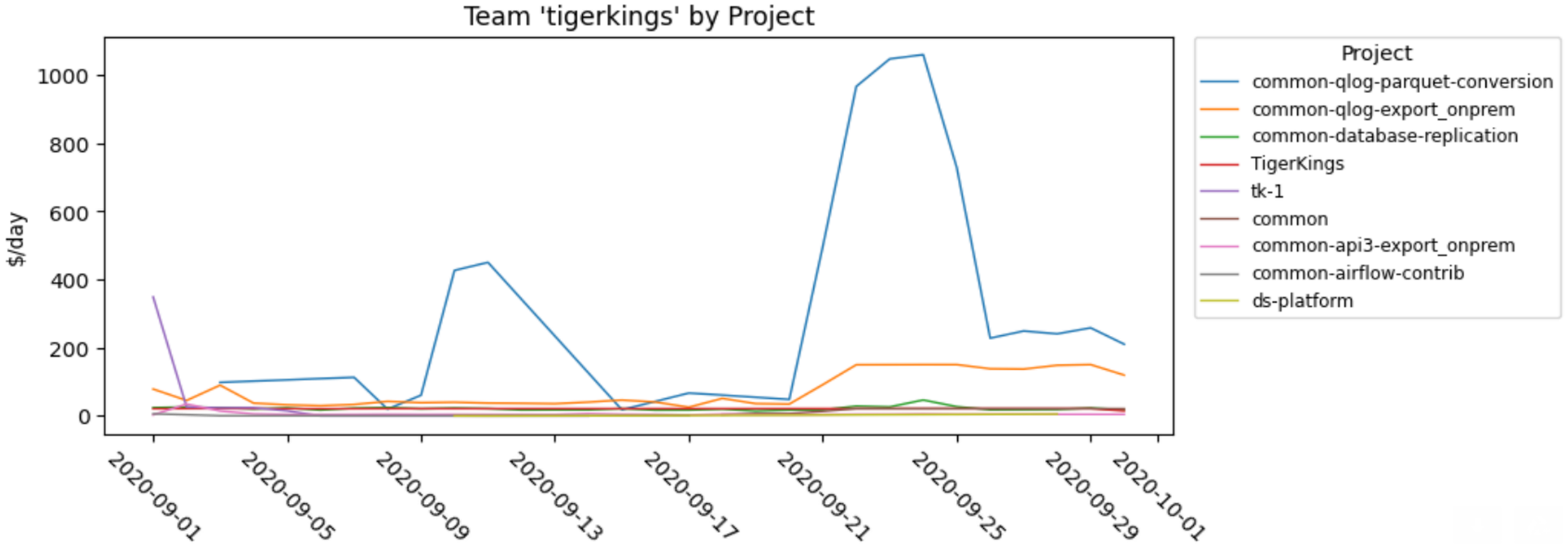
In practice, we save even more than that because the unused RI/SP discounts are mostly absorbed by our less-predictable development and staging usage



Cost optimization

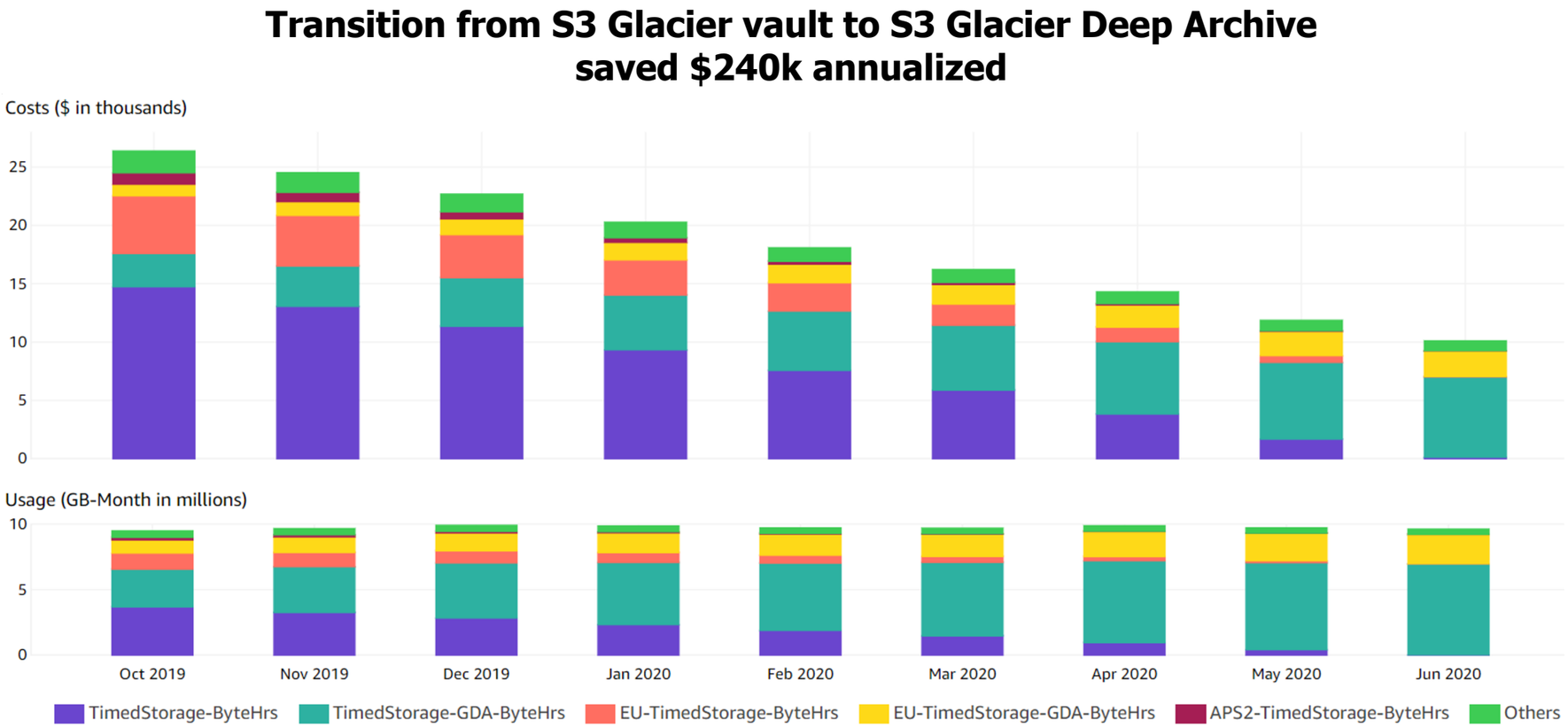
Get everyone involved

In the last 30 days, team 'tigerkings' has accrued $11,264 in AWS costs.
Here is a cost breakdown by project tag for the team:



Cost optimization

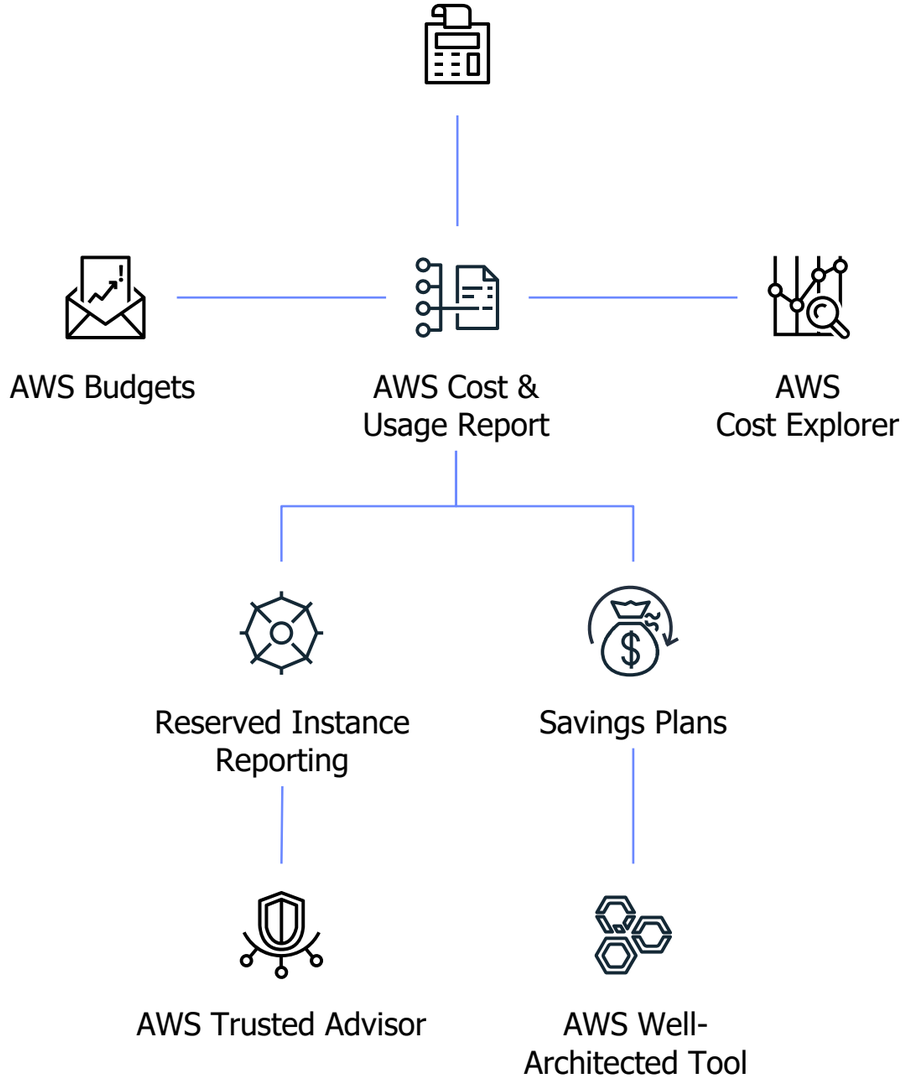
Leverage AWS resources to jump on new features



Cost optimization

Always be cost saving!

- Develop a FinOps mindset
- Schedule regular analysis
- Plan time for frequent cleanups
- Budget for migration development and testing
- Annualize your savings!



Focus is critical

Summary - Using AWS Cloud solutions and resources

Transitioning from collocations to AWS Cloud

Improved business continuity

Improved agility and engineering velocity

Amplified speed to market for product delivery

Drove scale and flexible infrastructure to meet demand

Improved performance for end users worldwide

Embrace FinOps – never stop optimizing for costs

Enriched acquisition integrations



Thank you!



AWS re:Invent

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



Please complete the session survey

AWS re:Invent

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