AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

You like...

Vacation at...

Movie: action or horror?

Bike or hike?

Early bird or night owl?

Pineapple!?

AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

AMZ303

Diving deep with Amazon Ads: Analytics at scale

Tom Skinner

Director,
Amazon Ads

Joshua Anghel

Principal Engineer,
Amazon Ads

Varun Kamalakaran

Principal, Customer Solutions
AWS

Agenda

Introduction

- Amazon advertising + Ads measurement team

Analytics at the core

- Analytics lifecycle
- Challenges, success measures, and key indications
- AWS collaboration

Technical solution – deep dive

- Big picture
- Architecture
  - Orchestration
  - Data applications
  - Reporting engines and APIs
- Lessons learned

amazon ads

amazon

alexa

freevee

music

fire tv

NFL

prime video

twitch

WHOLE FOODS MARKET

Amazon Ads Products

An array of advertising offerings to engage audiences and share the brand story

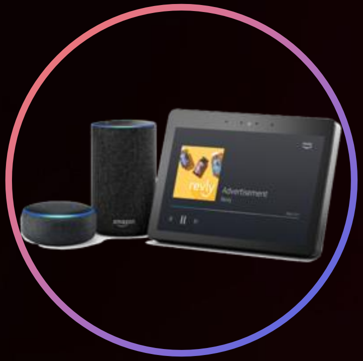
Sponsored Products

Sponsored Brands

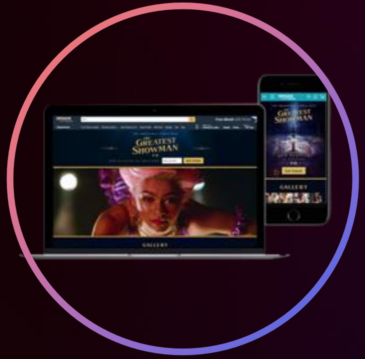
Sponsored Display

Display Ads

OOH Ads



Audio Ads



Custom Ads

Video Ads

amazon ads

↓

Ads measurement

100PB+
data

40B+
reports/year

400M
reports/day

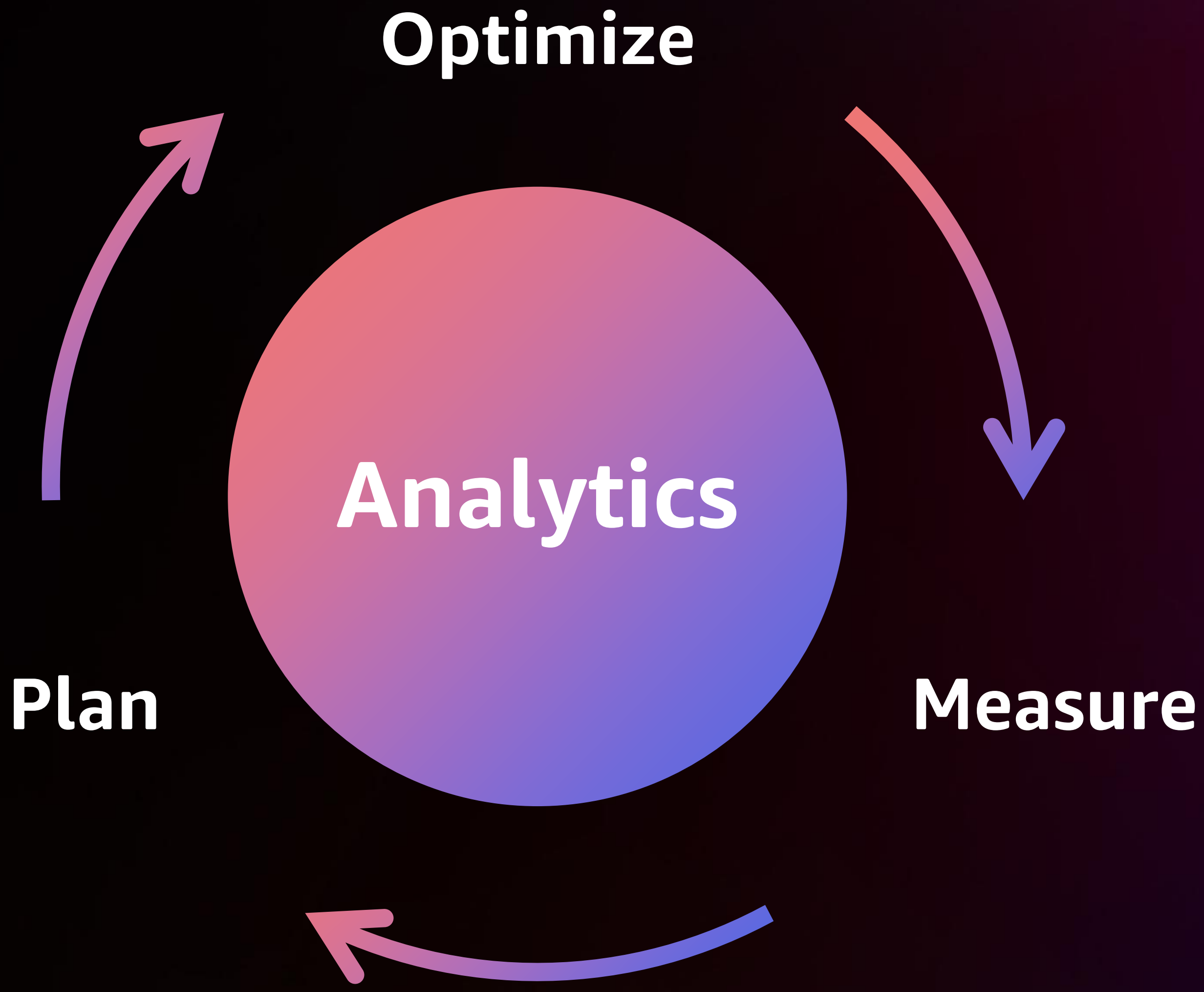
1T+
event/week

3M+
TPS

Analytics at the core

Tom Skinner
Director
Amazon Ads

Lifecycle—advertising analytics

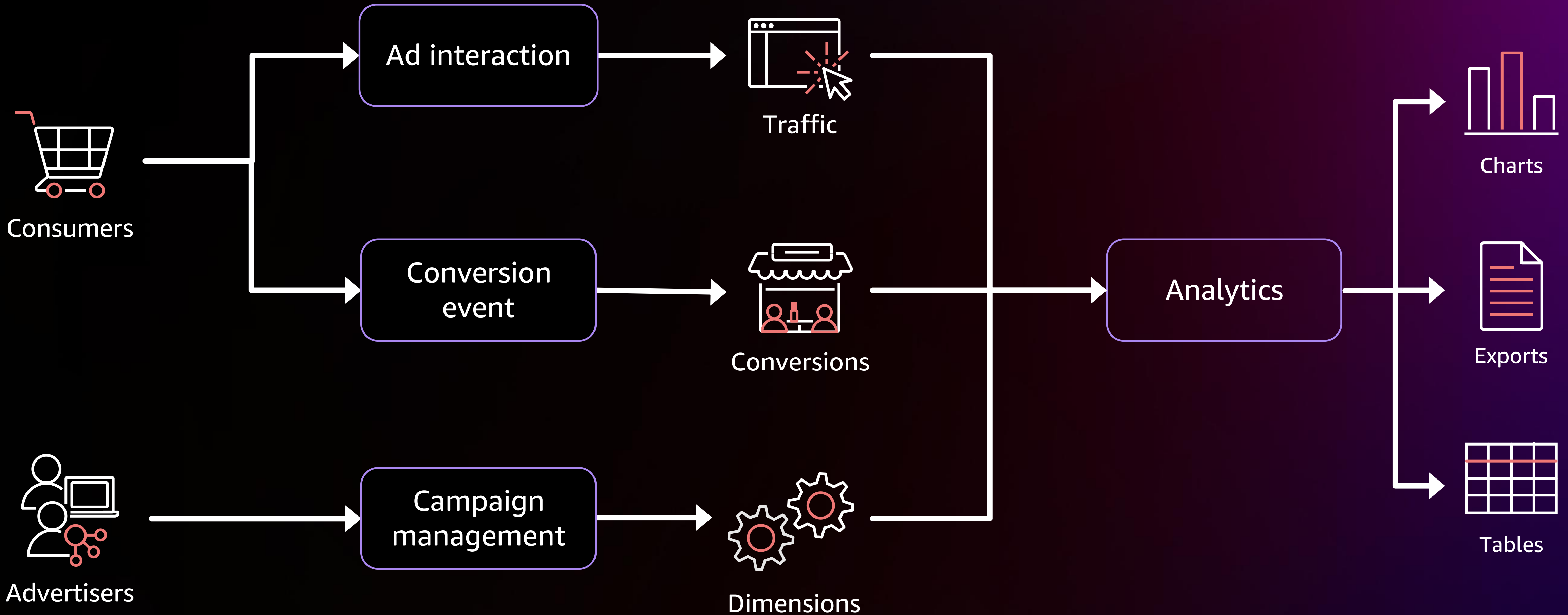

Optimize

Analytics

Plan

Measure

Data flows – Advertising analytics



Challenges

Critical to customer experience

Exactly once processing

Late arriving data

Long lookback calculations

Focus on the inputs, measure outputs

- Availability
- Operational load
- Cost efficiency
- Data quality
- Developer velocity
- Sustainability

Focus on the inputs, measure outputs

- Availability
- Operational load
- Cost efficiency
- Data quality
- Developer velocity
- Sustainability

Availability KPIs

Health check

Service availability

Query latency

Data age

Focus on the inputs, measure outputs

- Availability
- Data quality
- Operational load
- Developer velocity
- Cost efficiency
- Sustainability

Data quality KPIs

- Loss rate
- Measurement benchmarks
- Anomalies
- Customer tickets

Focus on the inputs, measure outputs

- Availability
- Operational load
- Cost efficiency
- Data quality
- Developer velocity
- Sustainability

Operational load KPIs

- Operational stories/engineer
- CI/CD pipeline metrics
- Ticket queue depth
- Ticket severity counts

Focus on the inputs, measure outputs

Availability

Data quality

Operational load

Developer velocity

Cost efficiency

Sustainability

Developer velocity KPIs

Complexity

Automation percentage

Developer days

Focus on the inputs, measure outputs

- Availability
- Operational load
- Cost efficiency
- Data quality
- Developer velocity
- Sustainability

Cost efficiency KPIs

- System utilization
- Year-over-year trends
- Data processing cost
- Cost/user action

Focus on the inputs, measure outputs

- Availability
- Operational load
- Cost efficiency
- Data quality
- Developer velocity
- Sustainability

Sustainability KPIs

Instance type selection

Region selection

Carbon footprint

Advertising – AWS services

Amazon EBS

Amazon S3

Amazon ECR

Amazon EKS

Amazon ECS

AWS Fargate

Amazon EC2

AWS Lambda

Amazon SNS

Amazon SQS

AWS Step Functions

AWS Glue

Amazon Kinesis

Amazon EMR

Amazon Athena

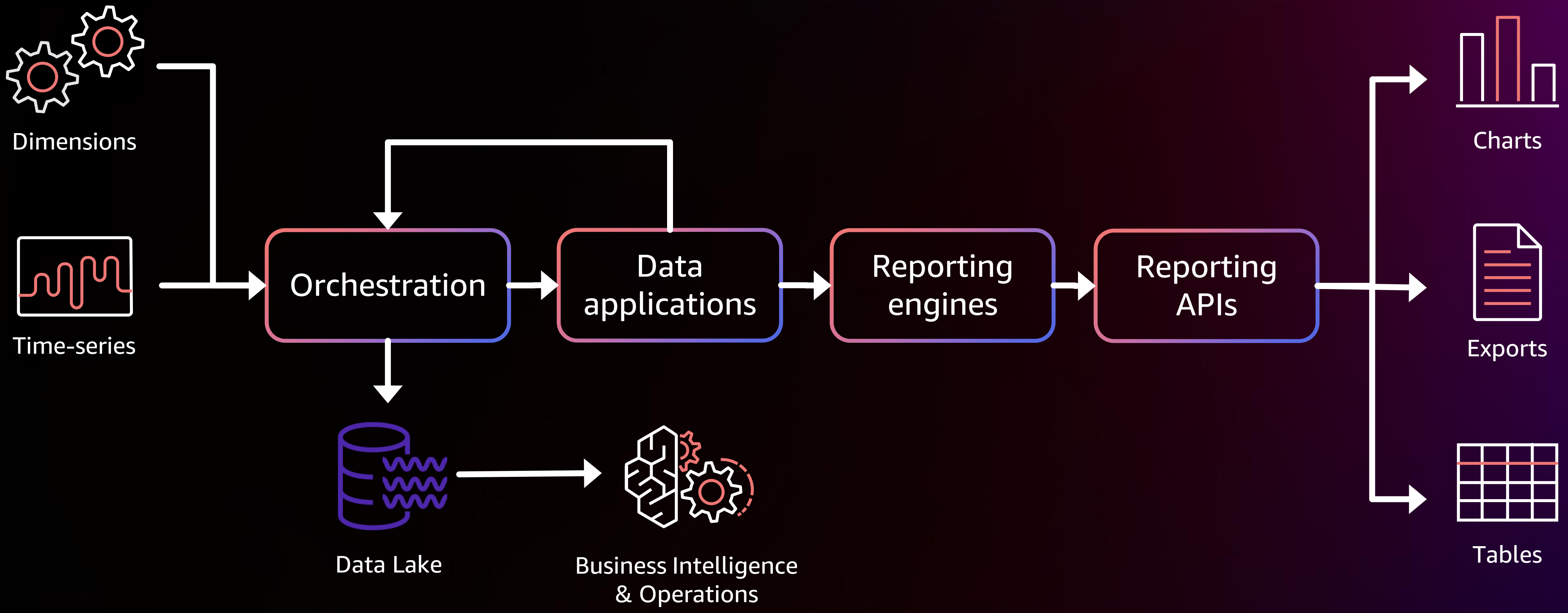
Amazon DynamoDB

Amazon CloudWatch

Technical solution

Josh Anghel
Principal Engineer
Amazon Ads

Big picture – Analytics



Dimensions

Time-series

Orchestration

Data applications

Reporting engines

Reporting APIs

Charts

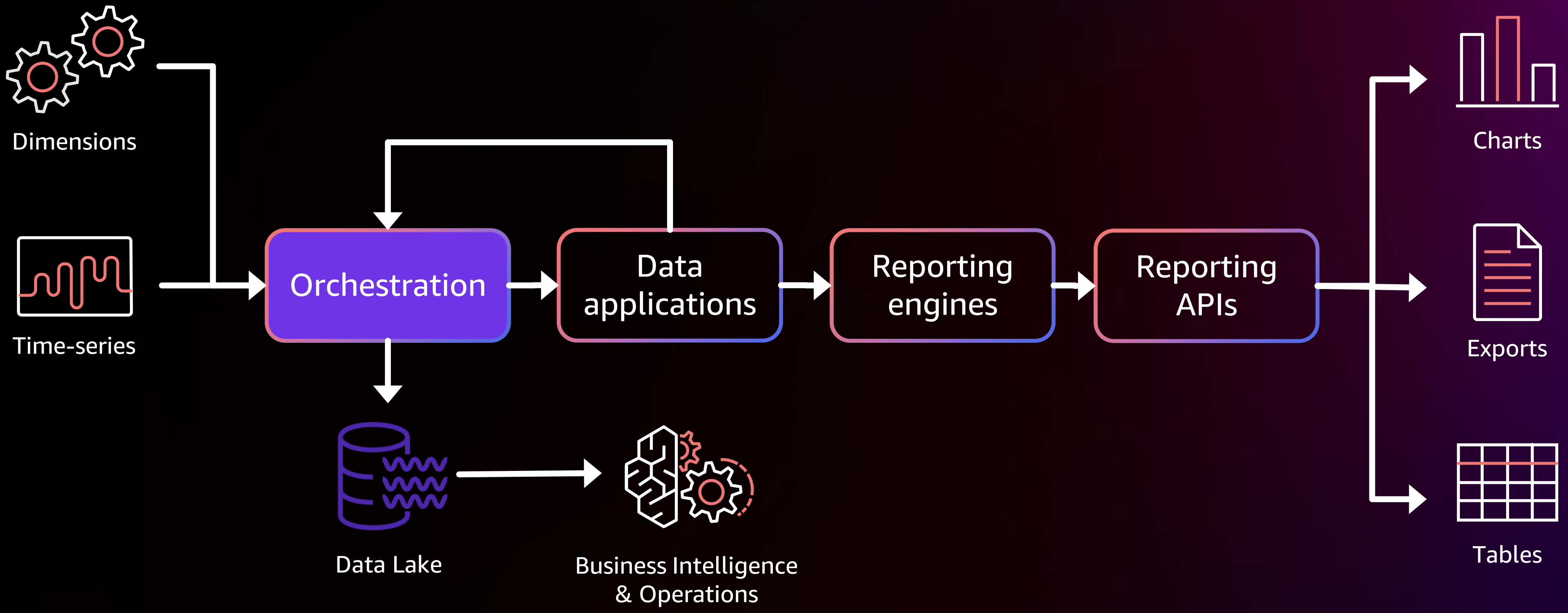
Exports

Tables

Data Lake

Business Intelligence & Operations

Orchestration



Orchestration

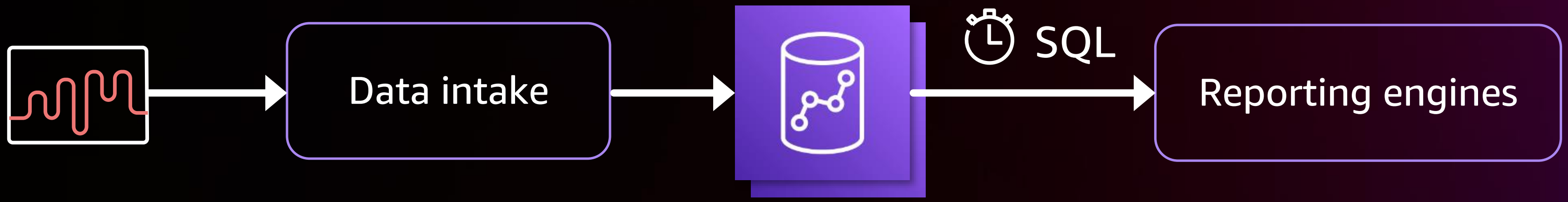
200
clusters

150k
jobs/day

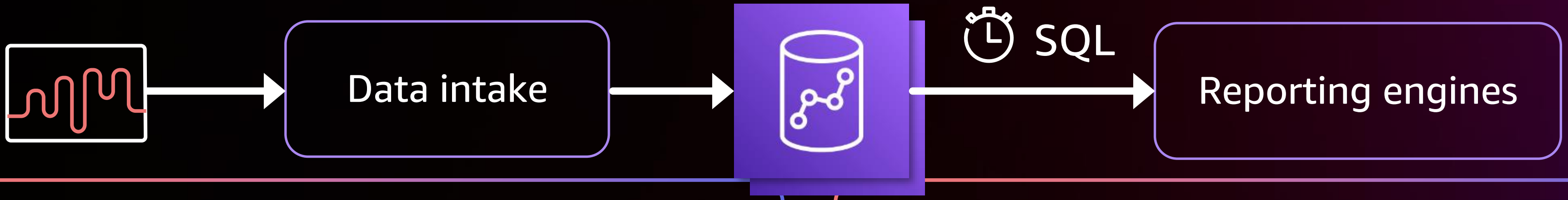
1T+
events/week

Exactly once

Orchestration V1 – Data warehouse



Orchestration V1 – Data warehouse



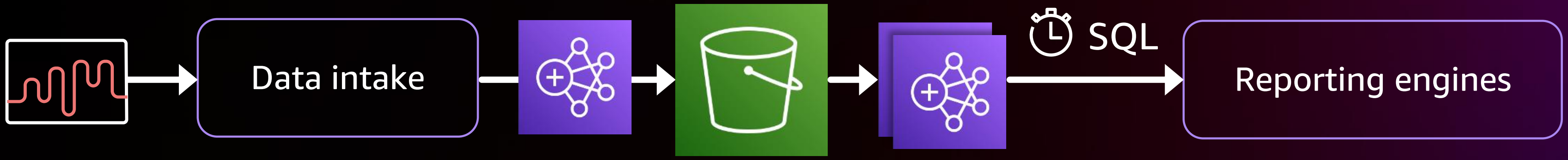
Data is stored on disk

Storage and compute are coupled

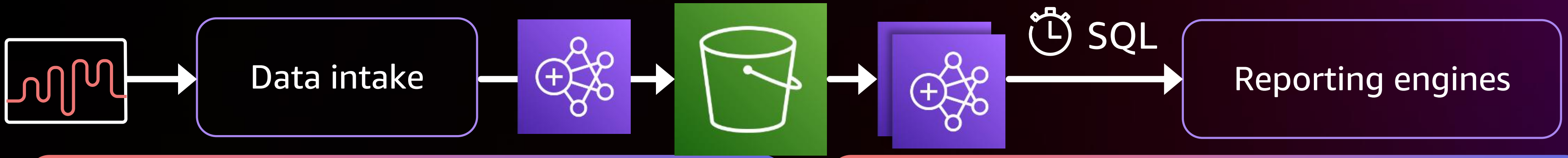
Source of record was unclear

Large monolithic clusters

Orchestration V2 – Data lake



Orchestration V2 – Data lake



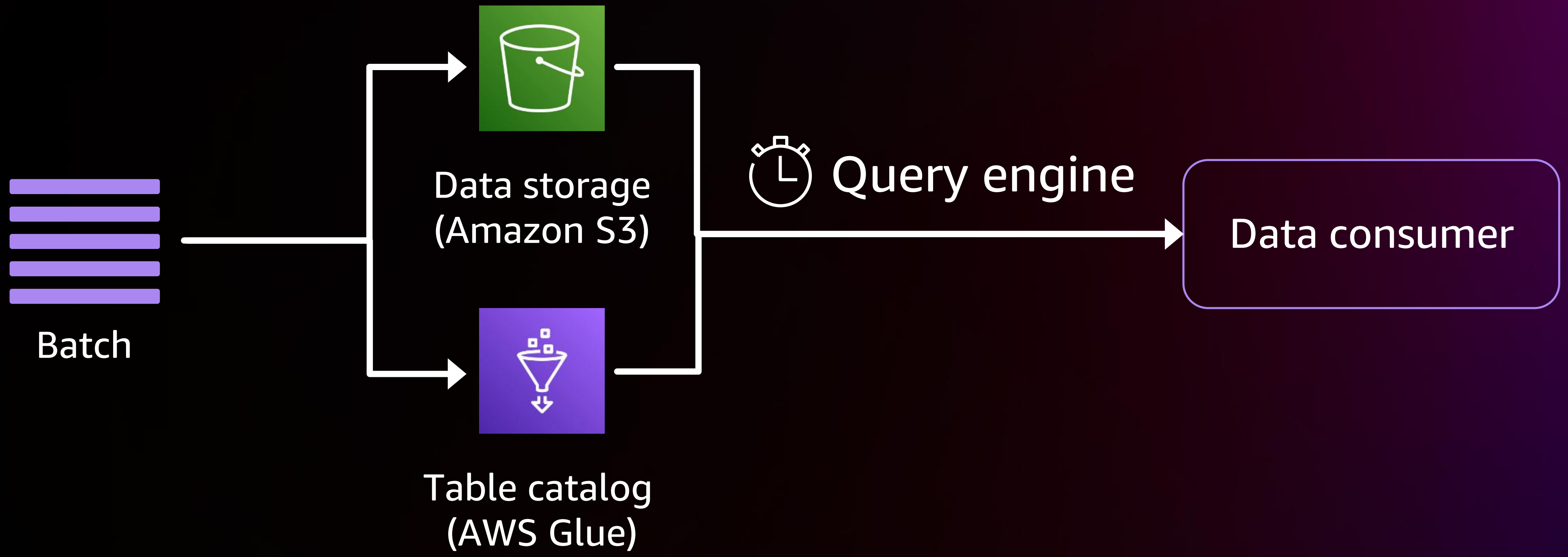
Data is stored in Amazon S3

Single source of record

Separate storage & compute

Optimal cluster sizes

Data lakes



Batch

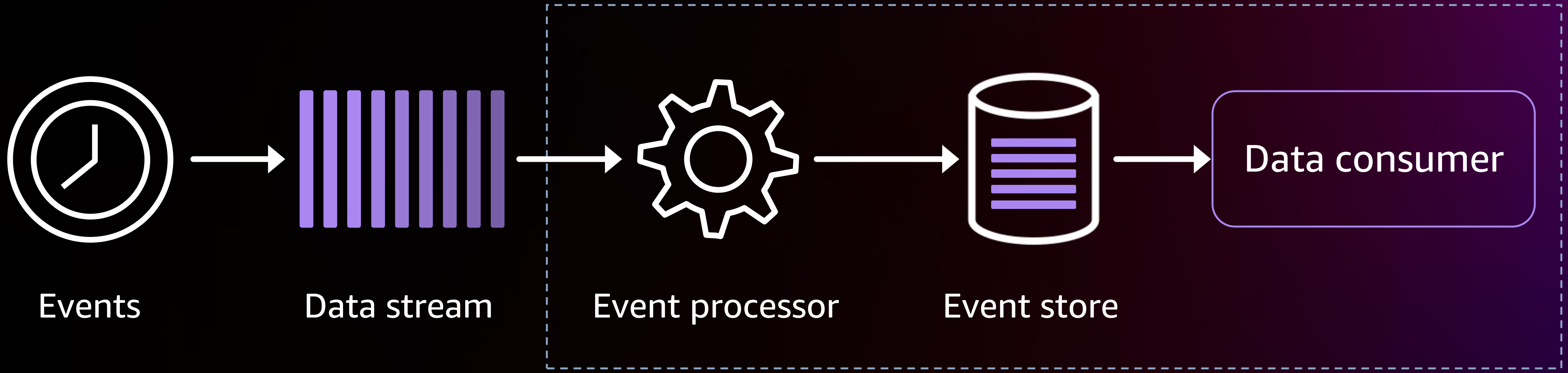
Data storage
(Amazon S3)

Table catalog
(AWS Glue)

Query engine

Data consumer

Data streams



Data streams – Limitations

- Event state storage
- Fault tolerance
- Cold-starts and backfills
- Developer velocity
- Lookbacks & aggregation
- On-demand scaling

Data streams – Limitations

- Event state storage
- Fault tolerance
- Cold-starts and backfills
- **Developer velocity**
- Lookbacks & aggregation
- On-demand scaling

Data streams – Limitations

Event state storage

Developer velocity

Fault tolerance

Lookbacks & aggregation

Cold-starts and backfills

On-demand scaling

Data streams – Limitations

- Event state storage
- Developer velocity
- Fault tolerance
- Lookbacks & aggregation
- Cold-starts and backfills
- On-demand scaling

Data streams – Limitations

- Event state storage
- Developer velocity
- Fault tolerance
- Lookbacks & aggregation
- Cold-starts and backfills
- On-demand scaling

Data streams – Limitations

Event state storage

Developer velocity

Fault tolerance

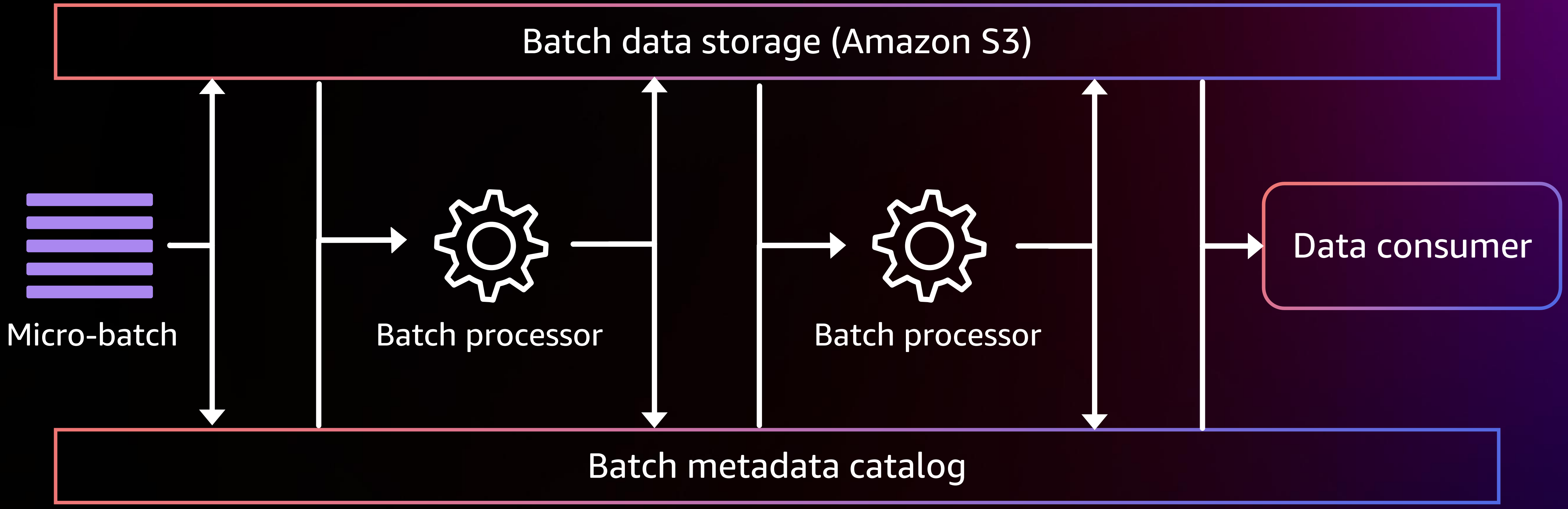
Lookbacks & aggregation

Cold-starts and backfills

On-demand scaling

Orchestration V3

Orchestration V3 – Data rivers

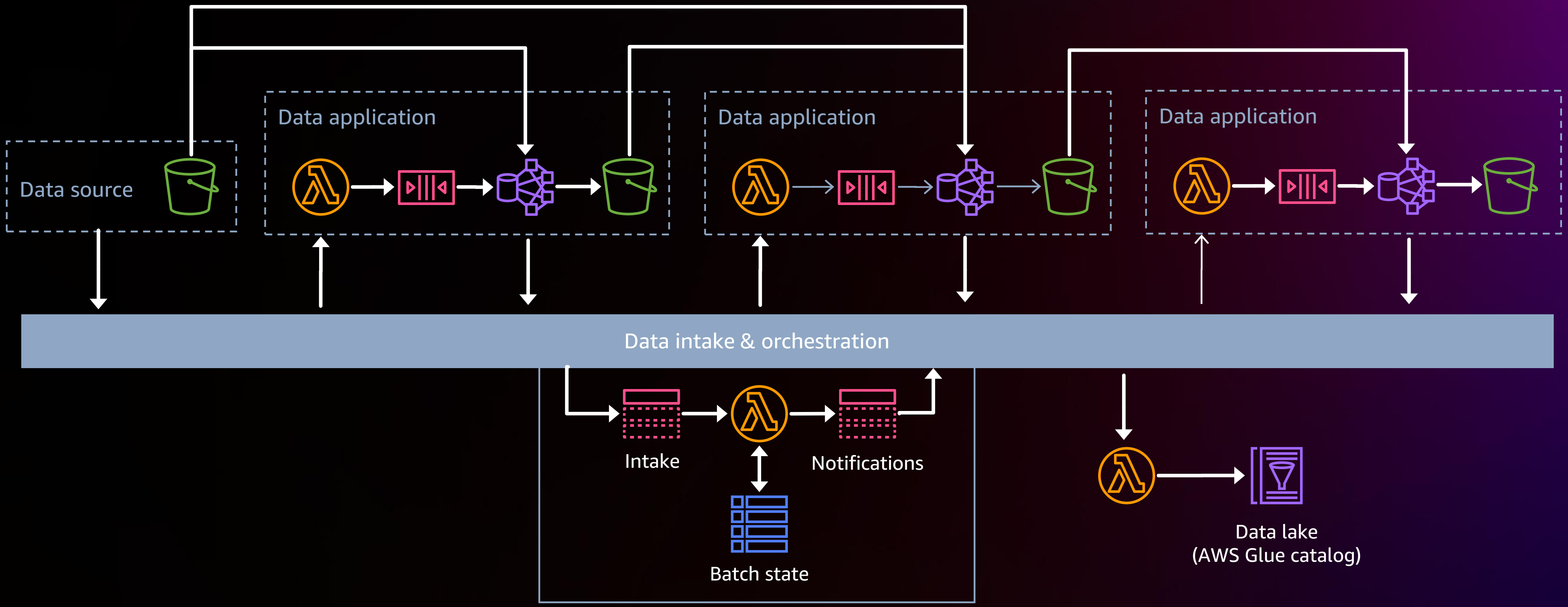


Orchestration V3 – Data rivers

	Data lake	Data river	Data stream
Data processing cost	★★★★★	★★★★	★★
Operational overhead	★★★★	★★★	★★
Developer velocity	★★★★	★★★★	★★
Data age	★	★★★★	★★★★★

Legend: more ★ indicates better results

Orchestration V3 – Data rivers



Data rivers – Outcome

Data availability

On-demand scaling

Continuous

Operational savings

Data processing cost savings

Orchestration – Lessons learned

Separate storage and compute

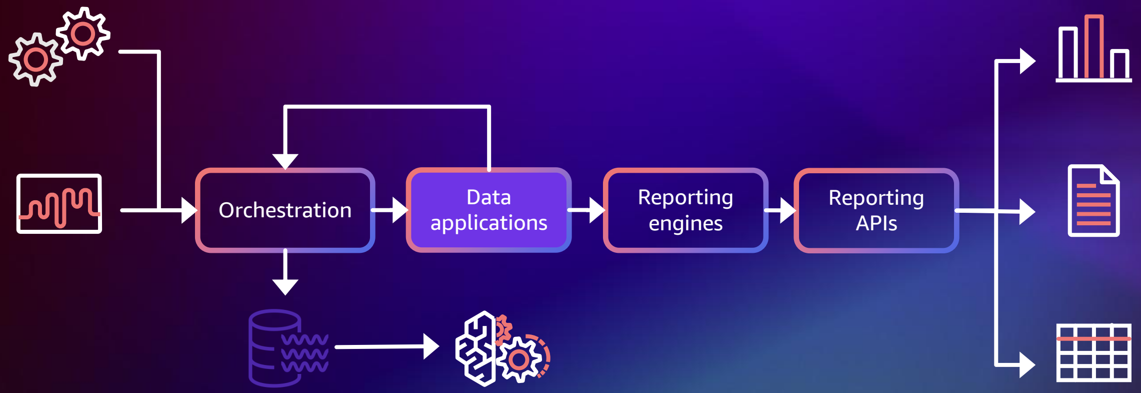
Avoid monolithic clusters

Choose the right technology

Process data in small batches

Data rivers offer a compromise between lake and stream

Data applications

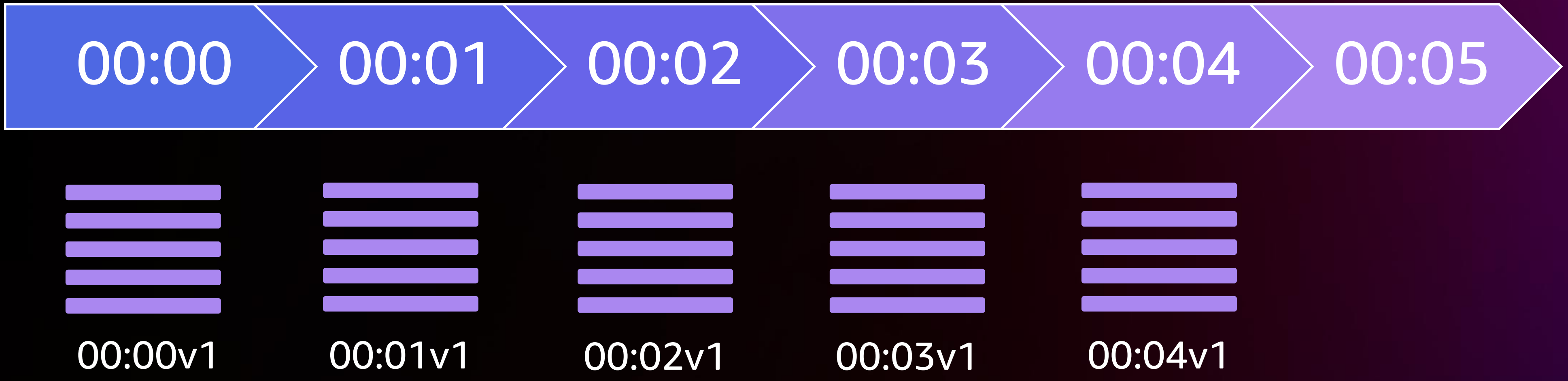


Data application

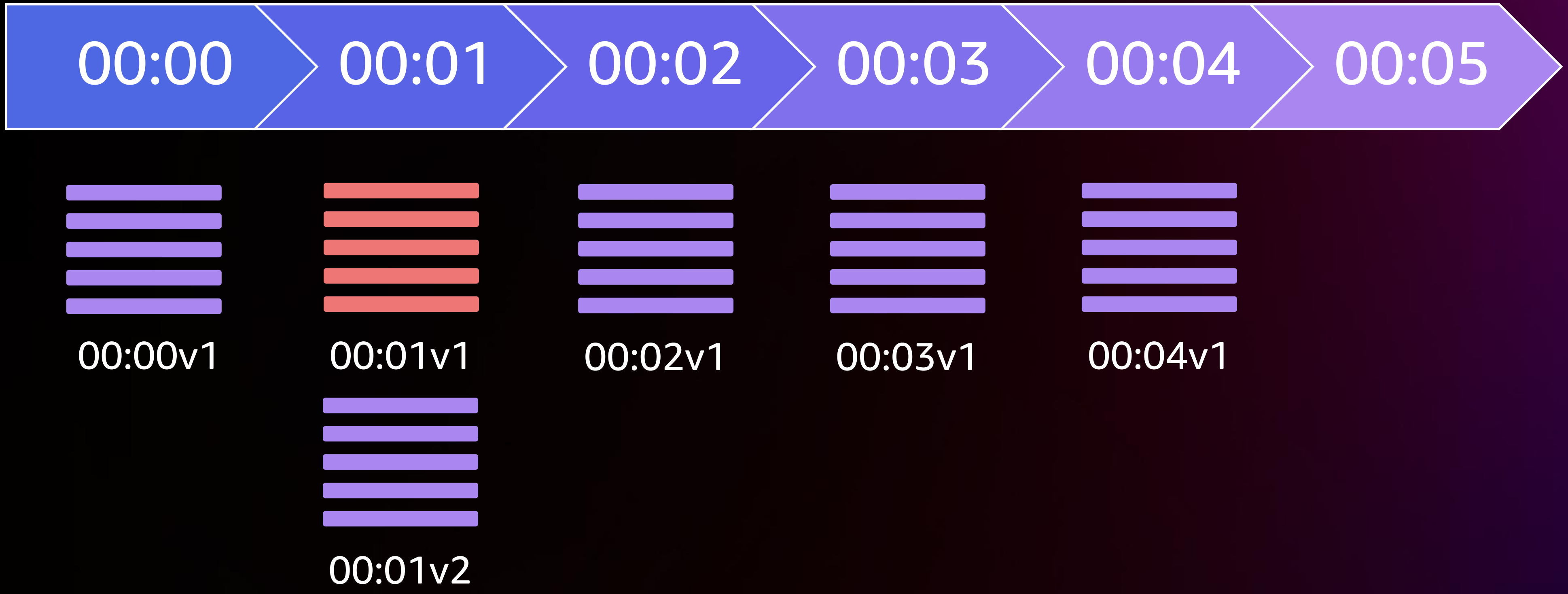
Partitioning

Aggregation

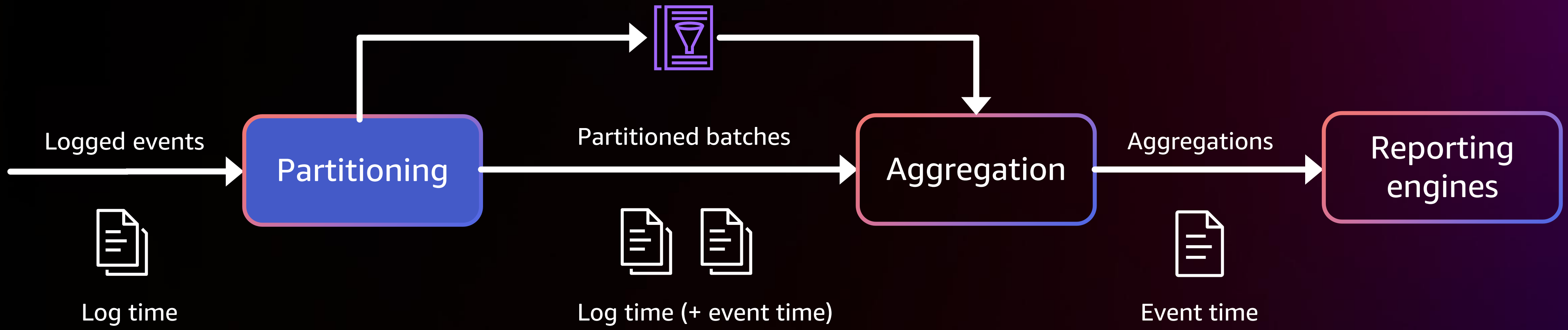
Batches


00:00
00:01
00:02
00:03
00:04
00:05
00:00v1
00:01v1
00:02v1
00:03v1
00:04v1

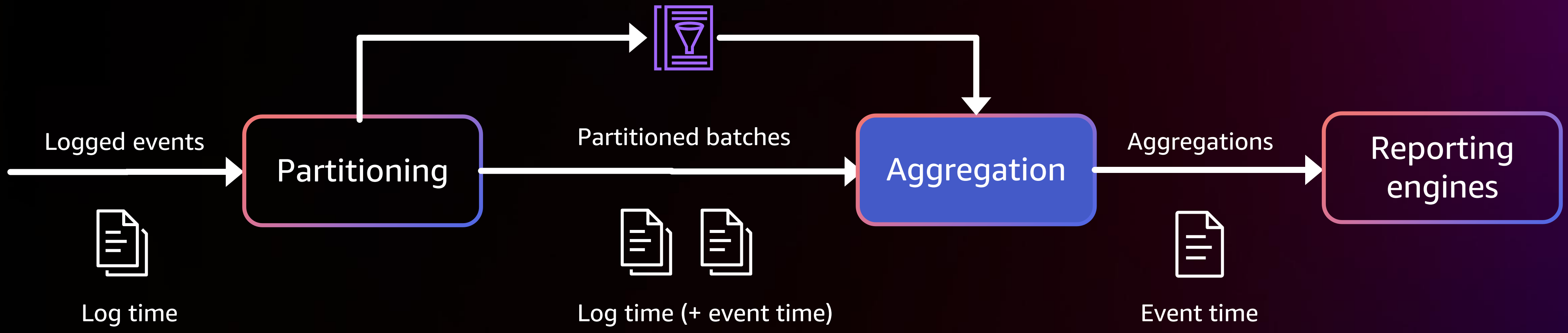
Restatements



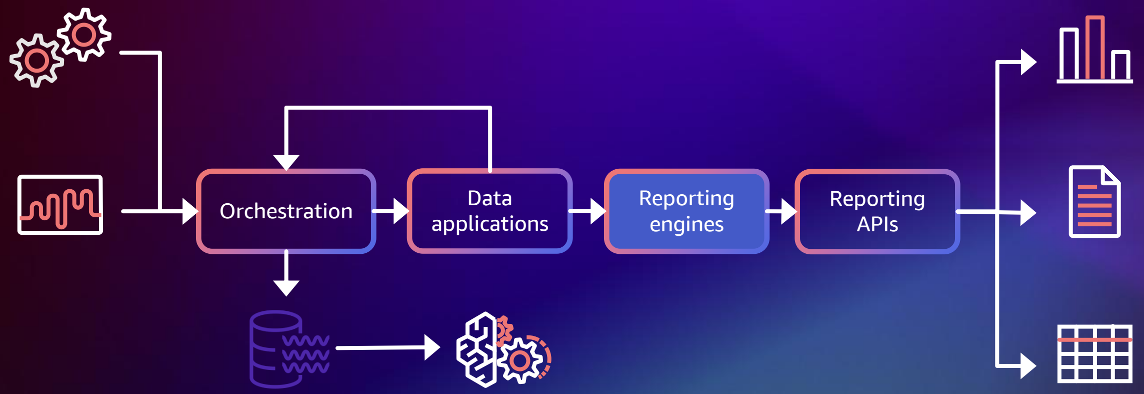
Data applications



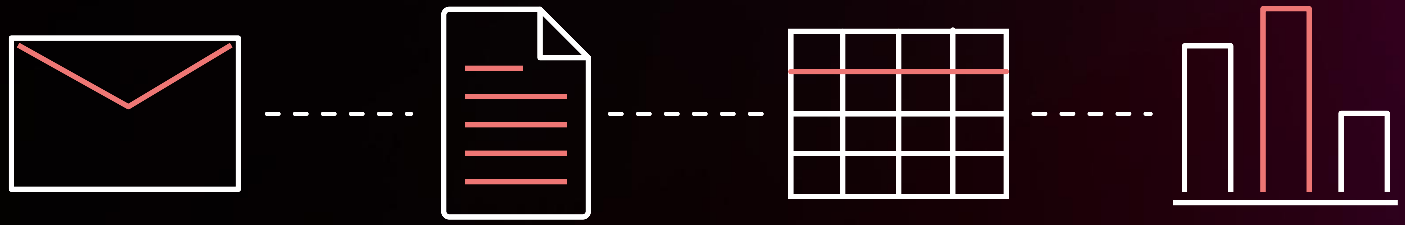
Data applications



Reporting engines



What is a reporting?



1.5M
indexed
per second

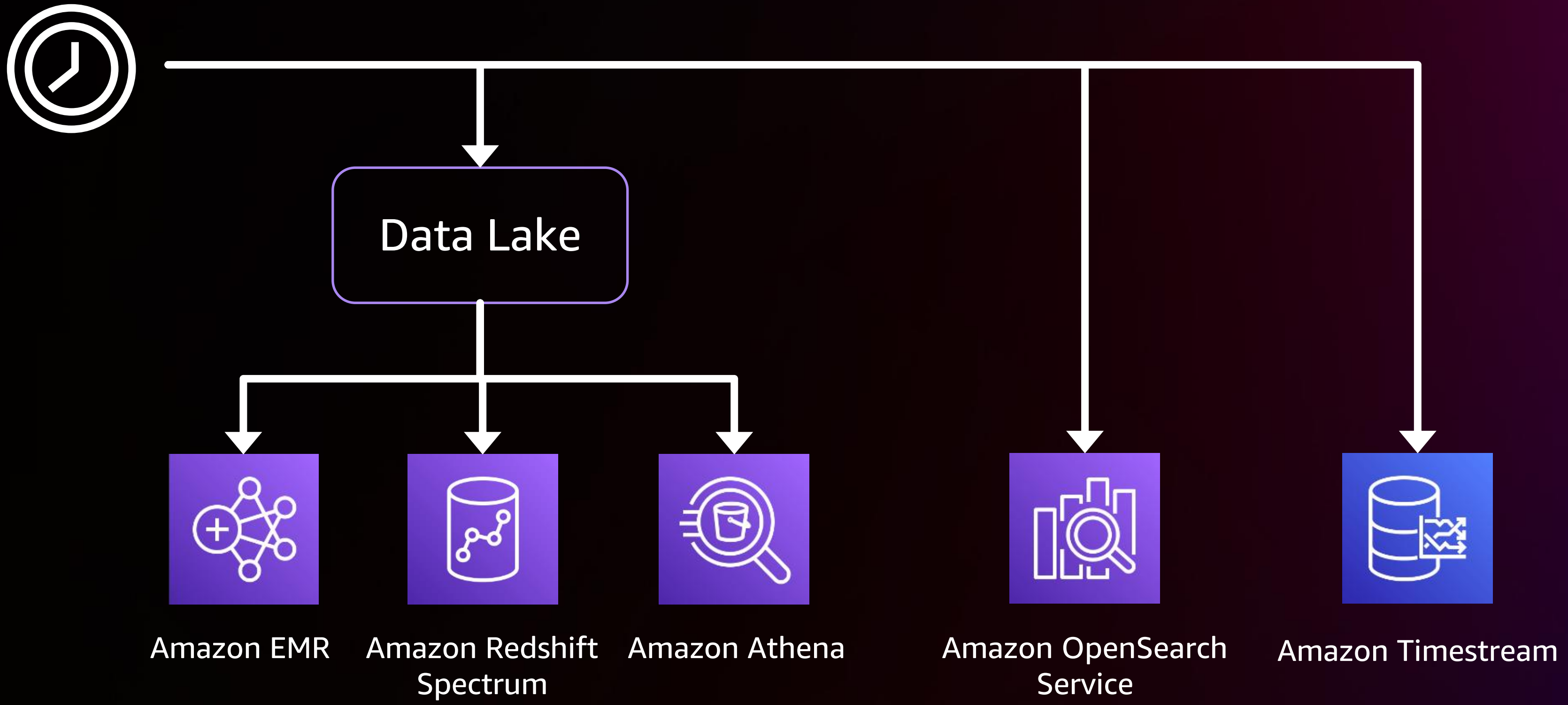
300M
scanned
per second

28M
aggregated
per second

8k
requests
per second

<2s
request
latency

Reporting engines



Data Lake

Amazon EMR

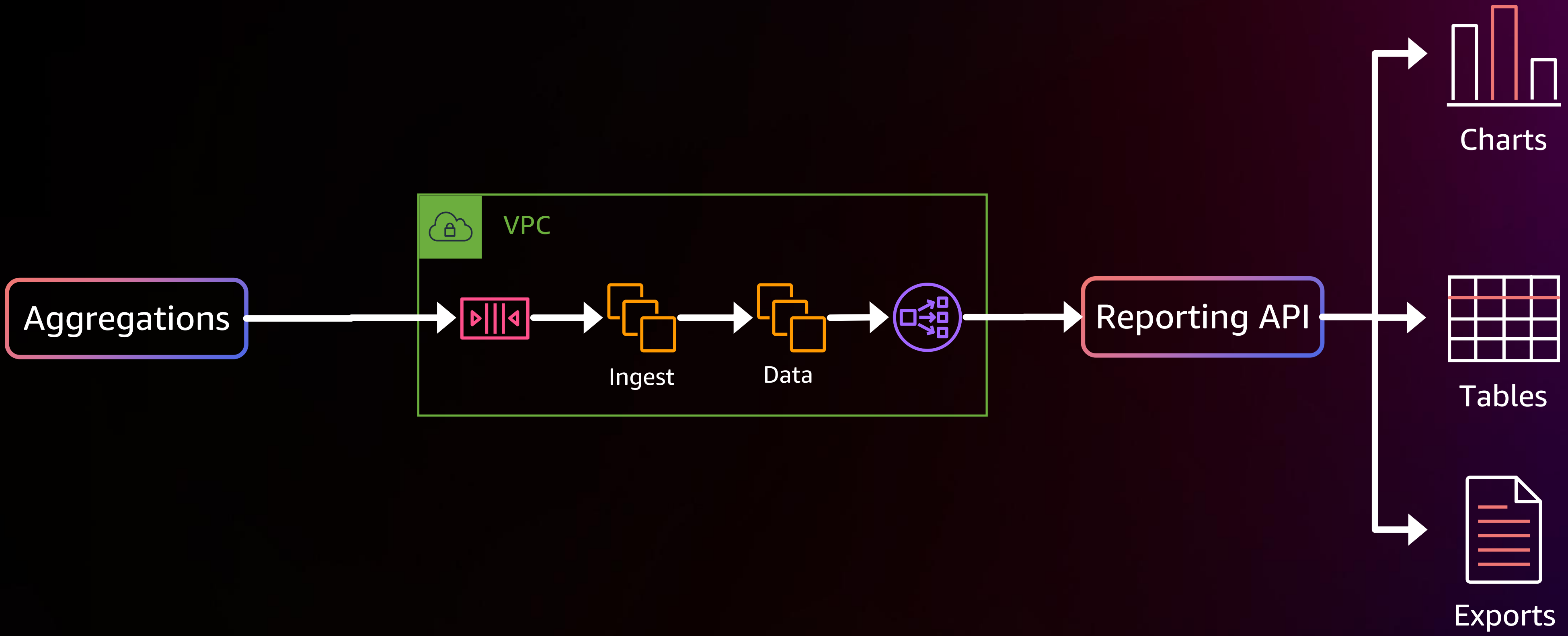
Amazon Redshift Spectrum

Amazon Athena

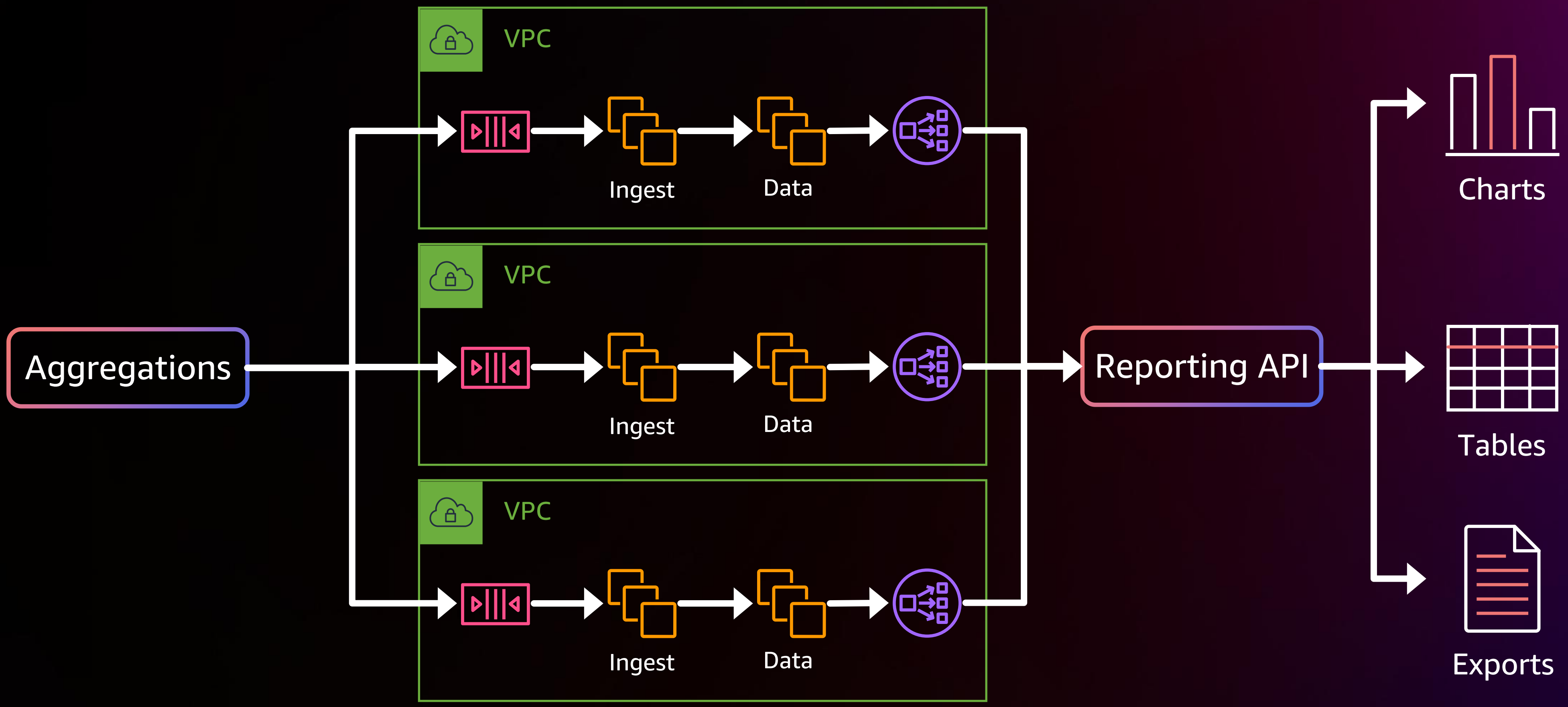
Amazon OpenSearch Service

Amazon Timestream

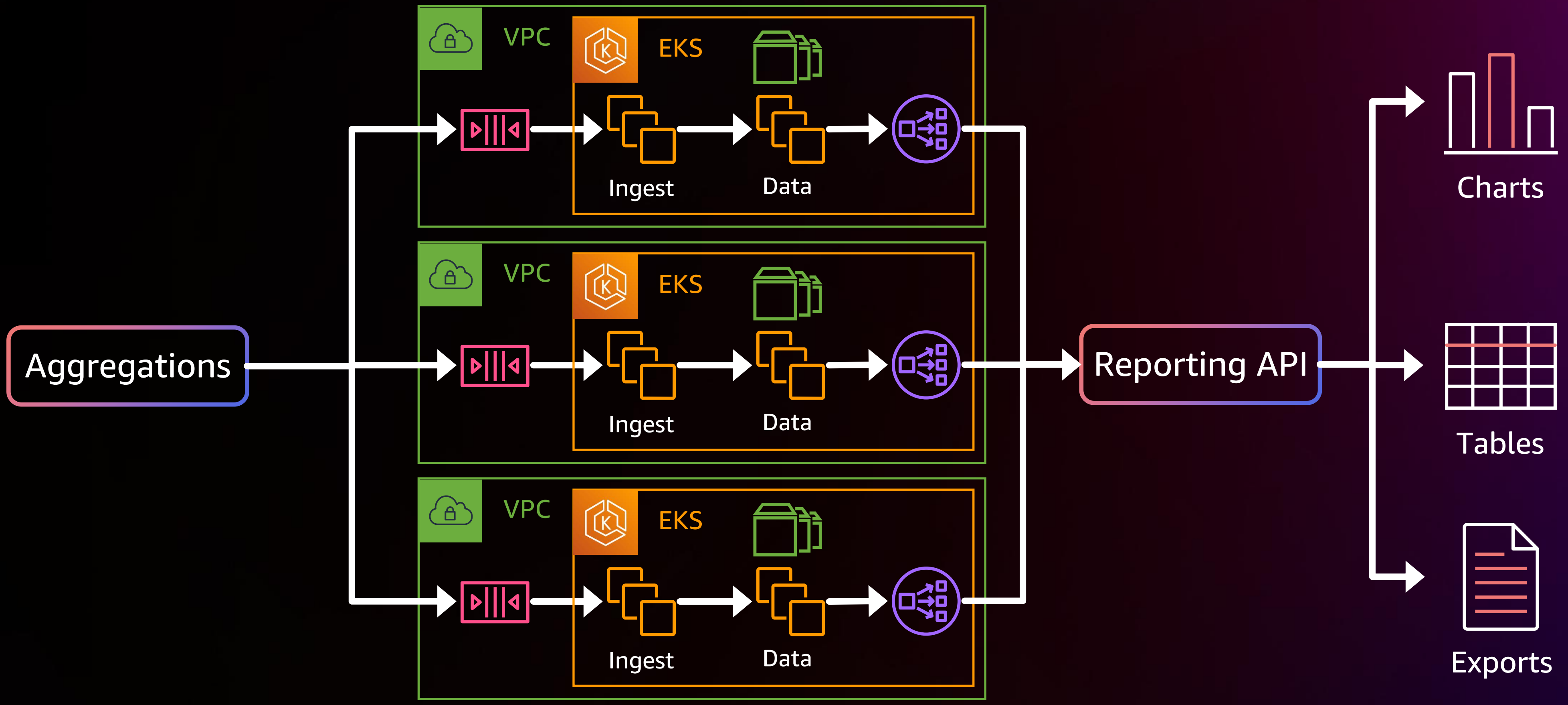
Reporting engines V1



Reporting engines V1



Reporting engines V2



Reporting engines – Lessons learned

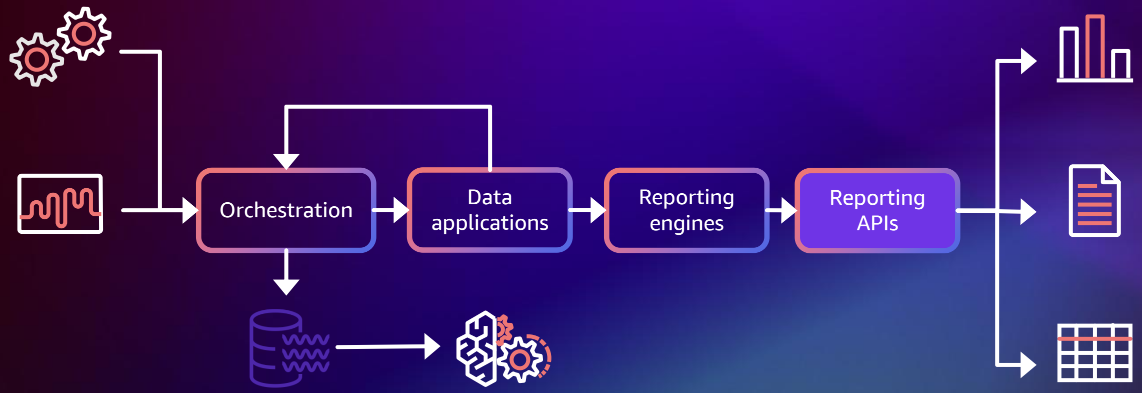
Separate storage and compute

Avoid monolithic clusters

Choose the right technology

Containers enable deploying multi-component services quickly

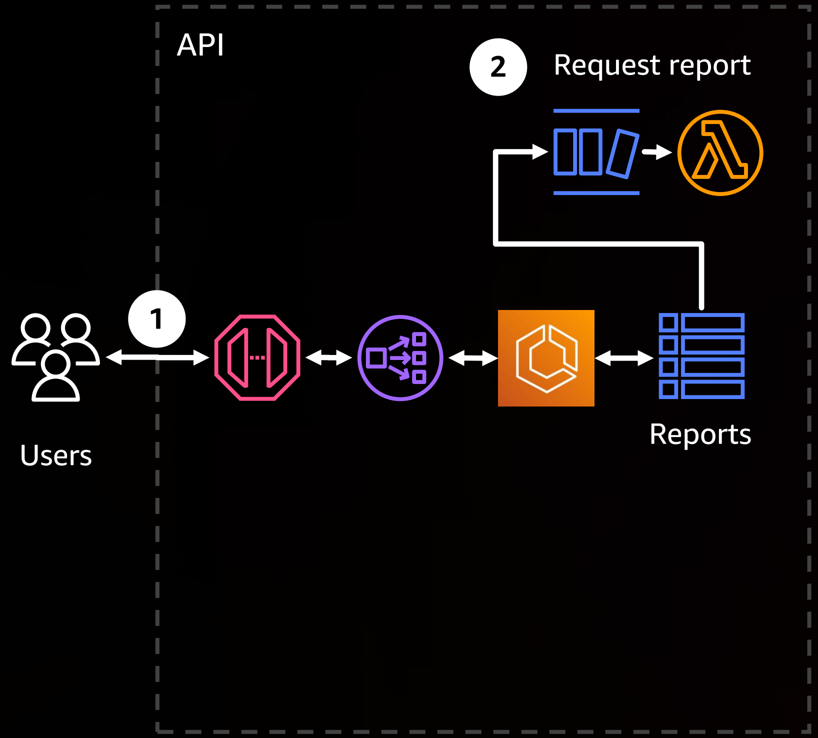
Reporting API



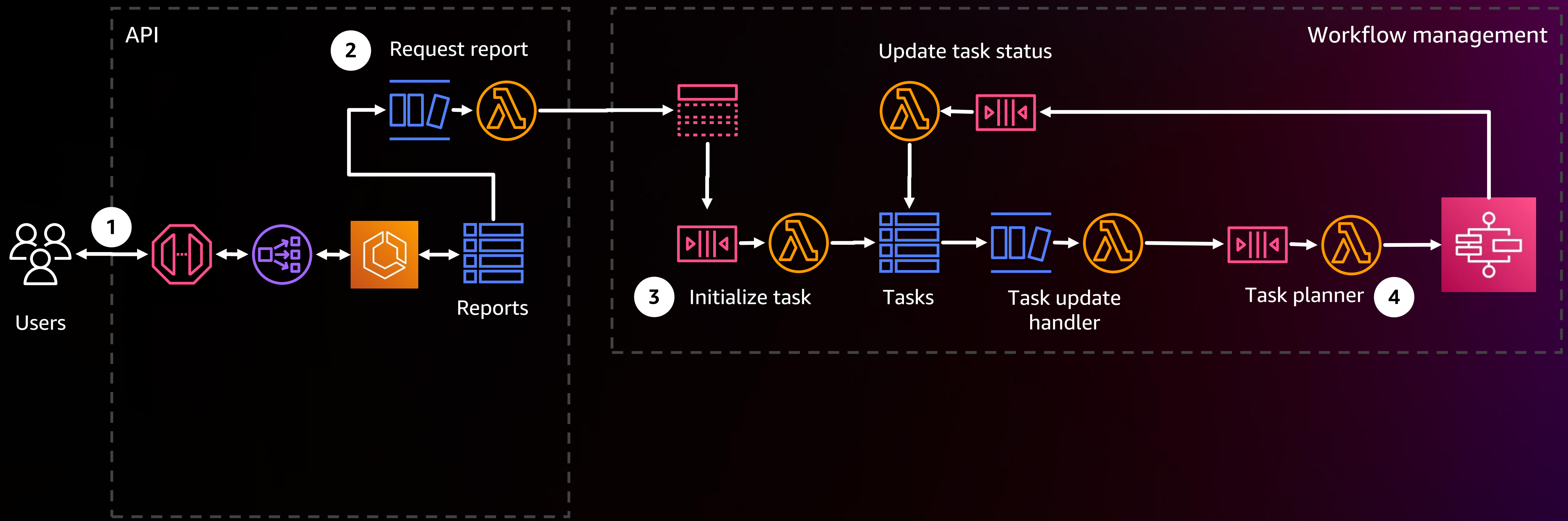
Reporting API – Exports

?

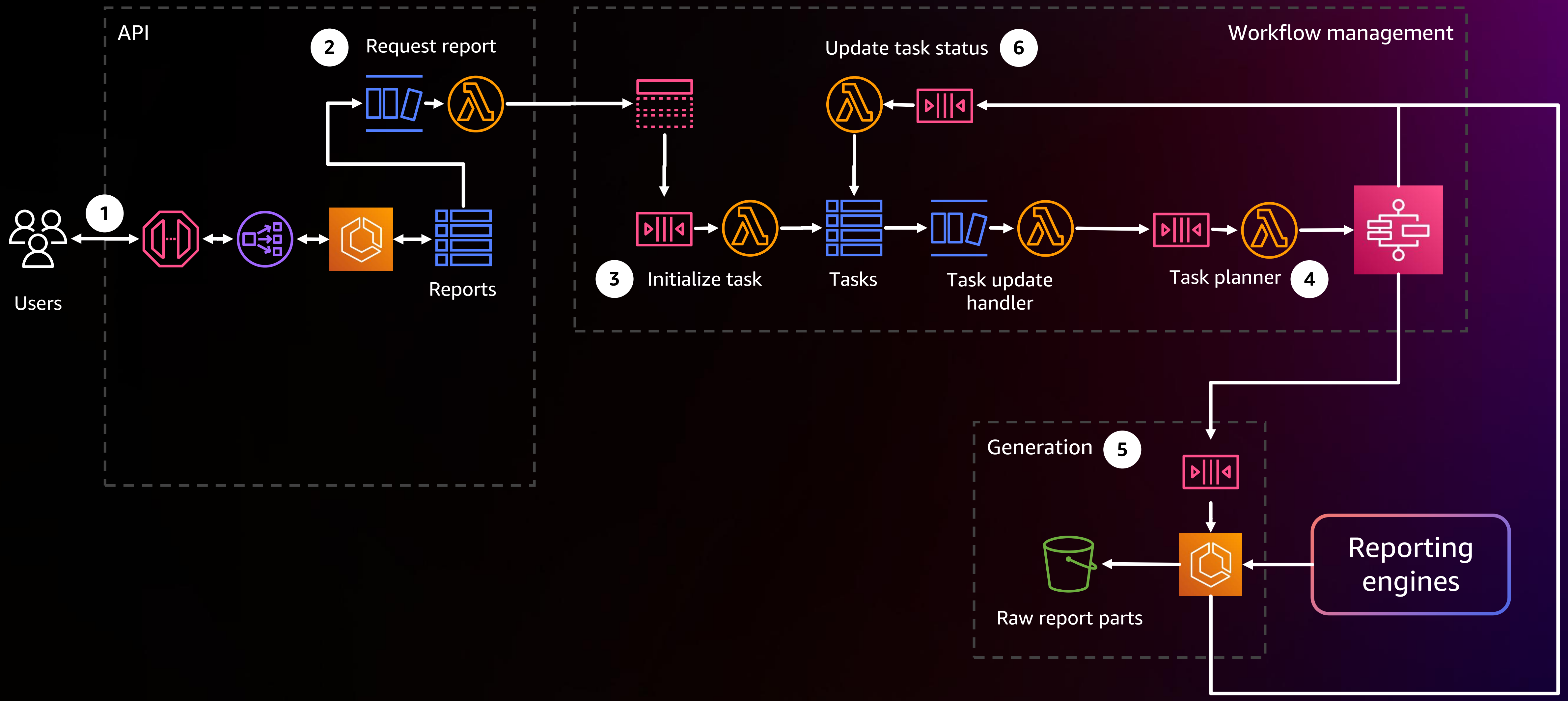
Reporting API – Exports



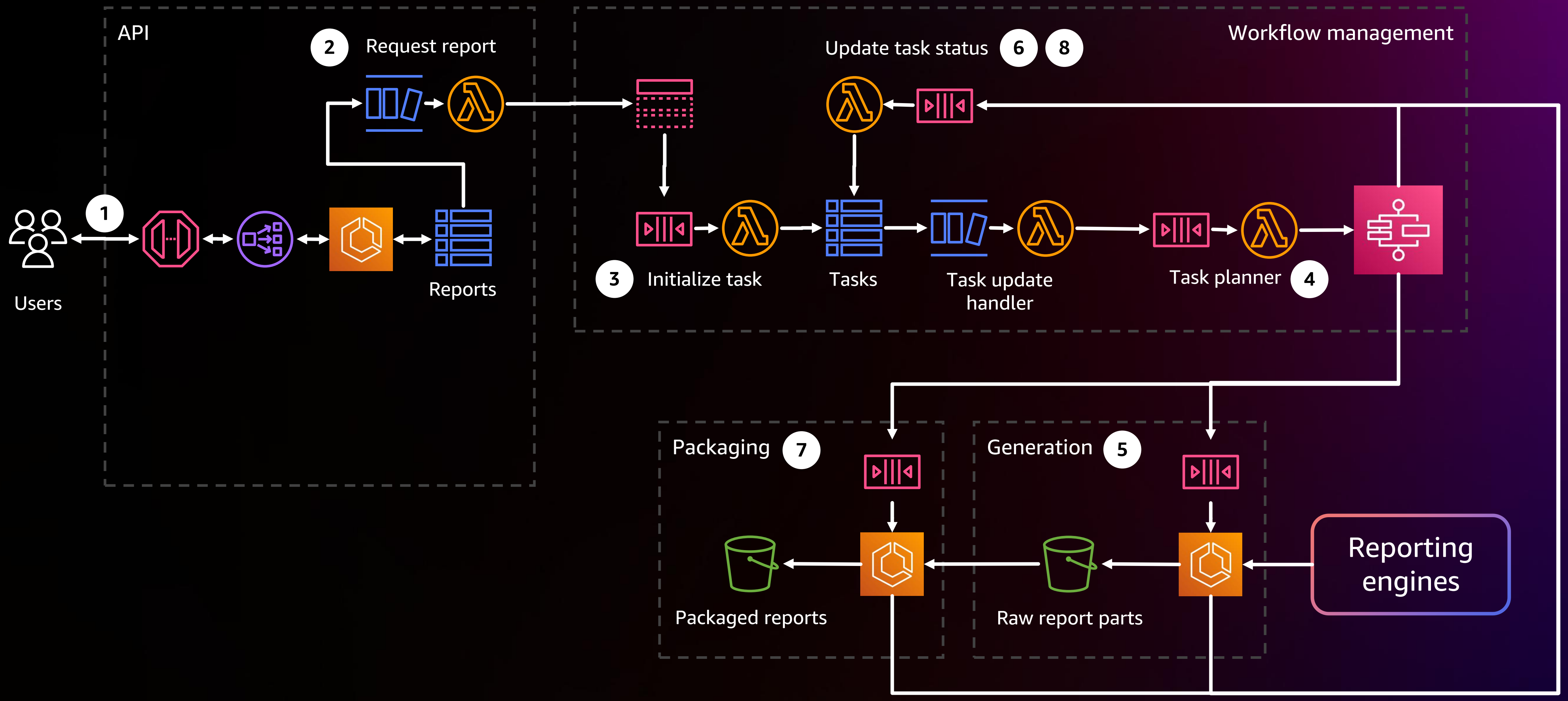
Reporting API – Exports



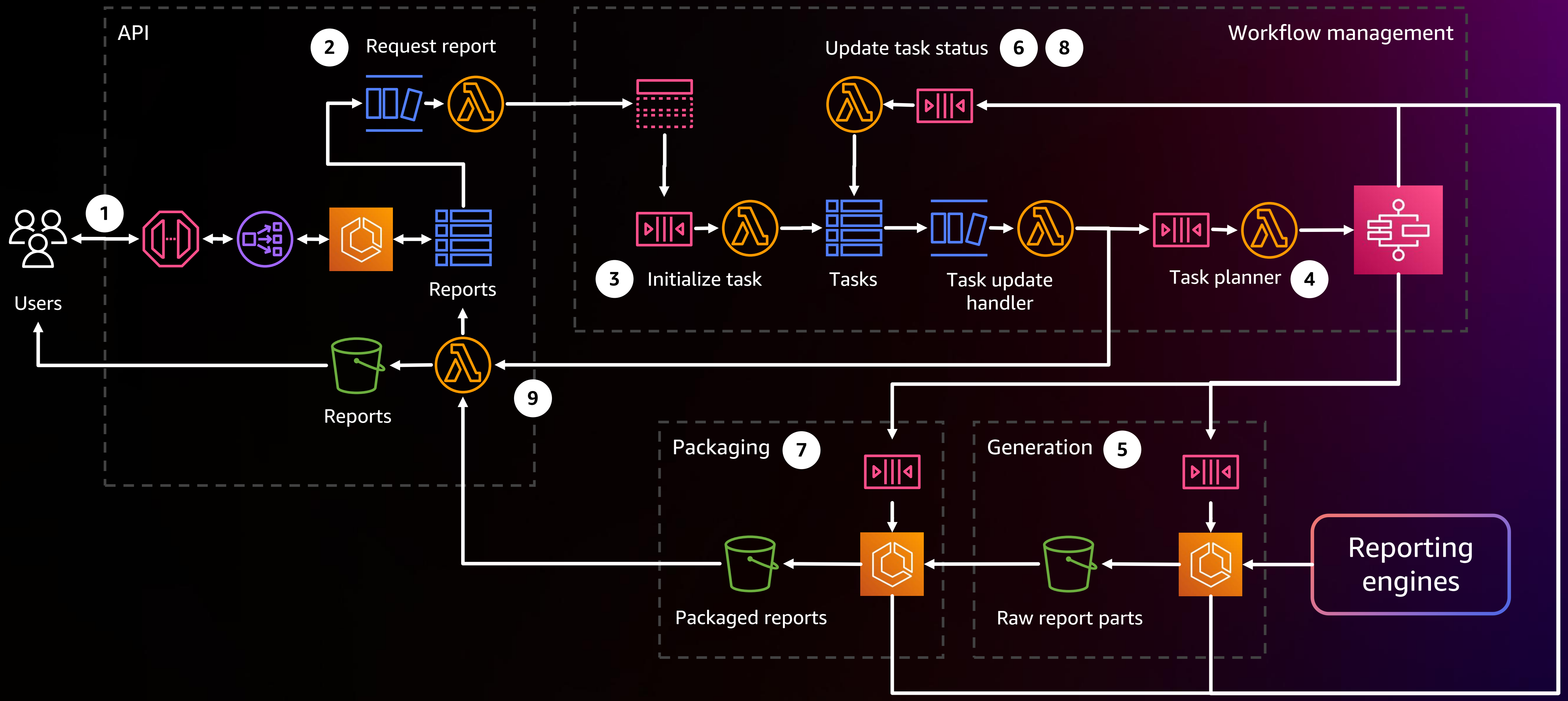
Reporting API – Exports



Reporting API – Exports



Reporting API – Exports



Reporting API – Lessons learned

Choose the right technology

Process data in small batches

Recap

- Orchestration with data rivers
- Partitioning and aggregating in a river
- Scaling high-TPS, low-latency reporting with Amazon EKS and OpenSearch
- Using step functions to build reporting exports incrementally

Thank you!

Amazon Ads booth: 2040 (behind AWS village)

Please complete the session survey in the **mobile app**