Lessons Learned Running Big Data on AWS

Michael Rhee, Architect – Gogo Data Warehouse and Analytics

August 1, 2018

Who we are

Gogo is the inflight internet company.

AVIATION-CENTRIC

RELENTLESSLY INNOVATIVE

PERFORMANCE OBSESSED

About Gogo

- 7,200+ Connected Aircraft
- 21 commercial airline partners
- Partnerships with the largest fractional ownership operators in business aviation
- 2,000+ aircraft awards for 2Ku, Gogo's latest commercial aviation technology
  - 19 airlines committed to 2Ku
- 120M+ connectivity sessions to date
- 140,000+ sessions/day
- 11,000+ flights/day

Gogo History at a Glance

1991

Gogo, previously known as the brand Aircell, begins in a BBQ restaurant in Denison, TX when founder Jimmy Ray sketches on a napkin a radical idea for an aircraft telephone system.

1997

Gogo begins providing phone service for business aviation.

2006

Gogo wins a major FCC auction: the only broadband frequencies allocated for communication between aircraft and ground-based cellular networks.

2011

Gogo Vision® officially launches on American Airlines, the first North American airline to offer the service.

2012

Gogo launches ATG-4, delivering triple the internet speeds of Gogo's current connectivity solution, ATG, to aircraft.

2014

Gogo announces 2Ku, its next-generation inflight connectivity technology.

2015

Gogo relocates its headquarters from Itasca, IL to downtown Chicago, IL.

2016

2Ku installed on more than 100 aircraft. Built backlog of 2Ku aircraft to beyond 1,500.

2017

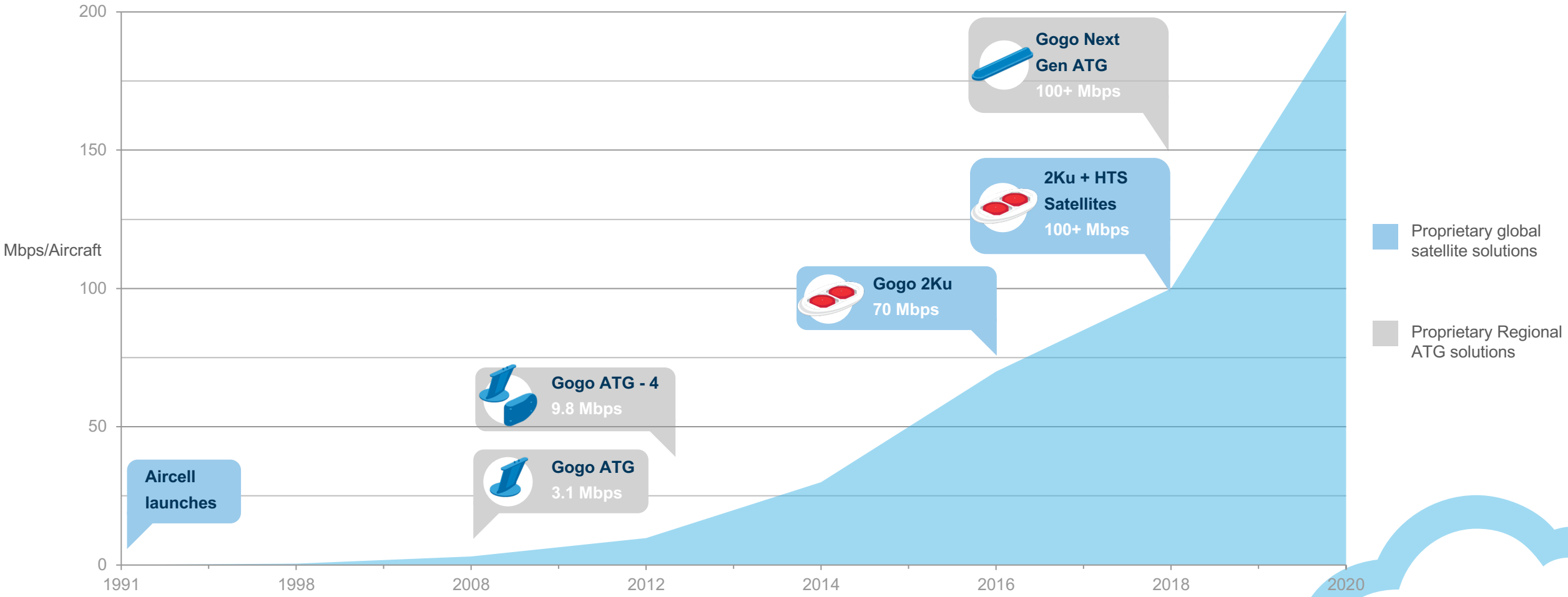
Gogo 2Ku technology hits new performance heights with 100+ Mbps during test flight.

Gogo named as supplier for Airbus.

1st 2Ku installation on an Airbus A350 Aircraft

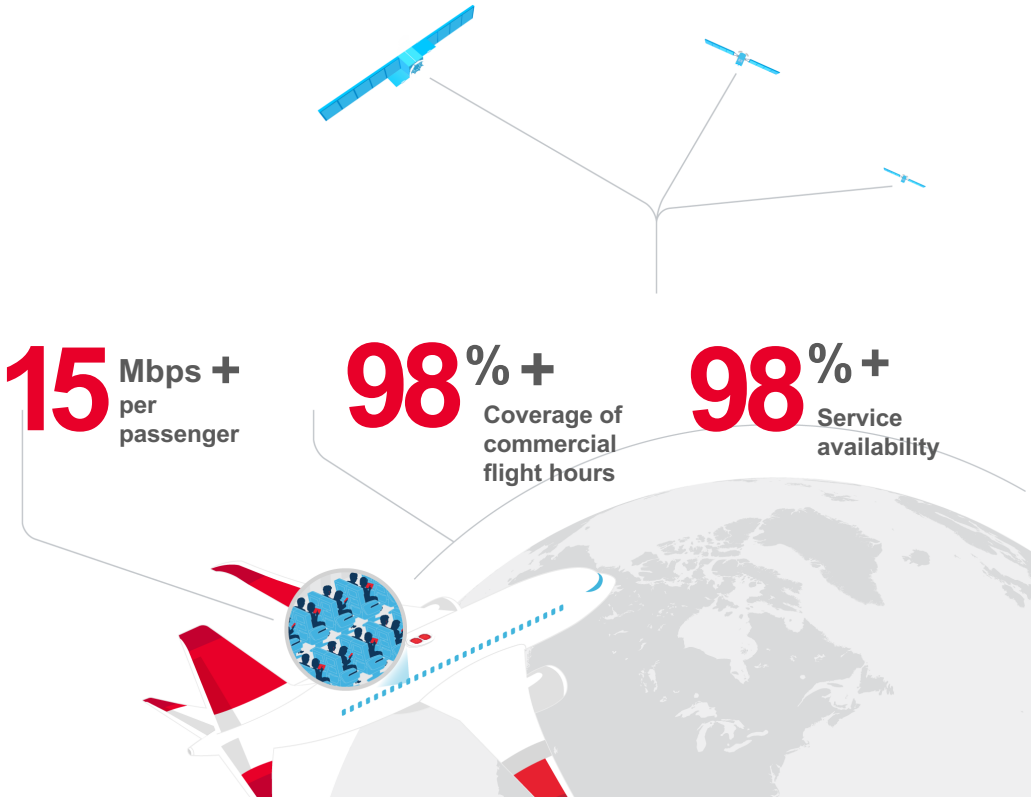
Alaska Airlines selects Gogo 2Ku for its Airbus fleet

Gogo's Technology Roadmap

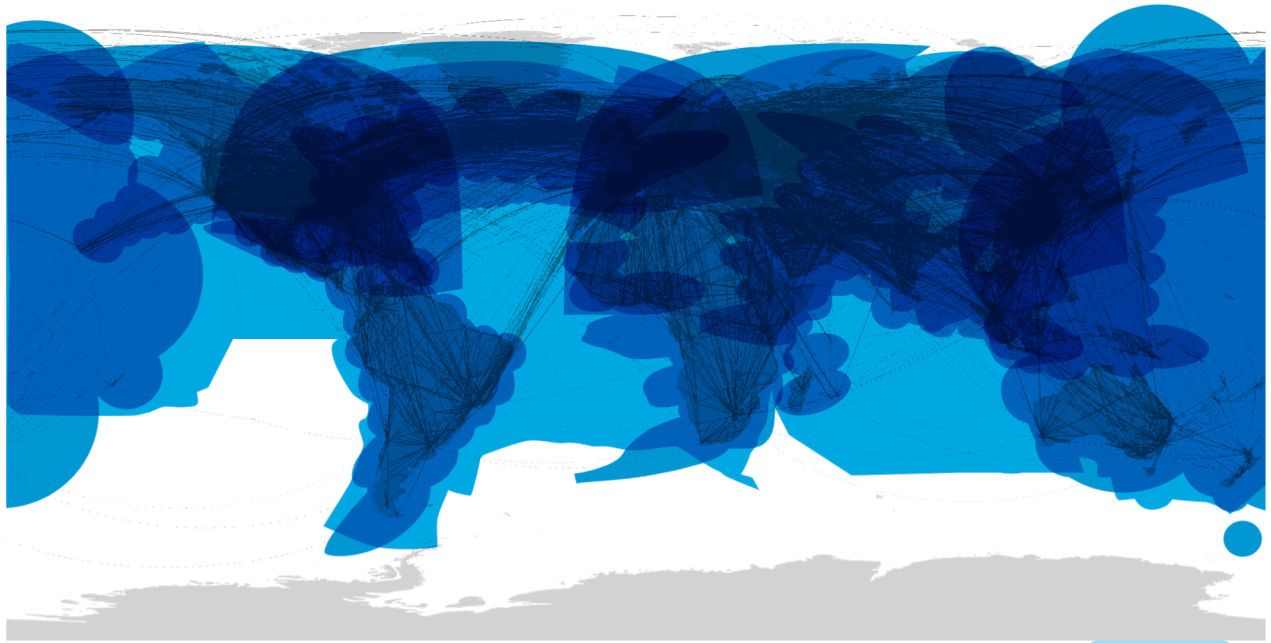


Performance everywhere with 2Ku

A ground-like experience



Everywhere aircraft fly



Backstory - Unified Data Platform (UDP)

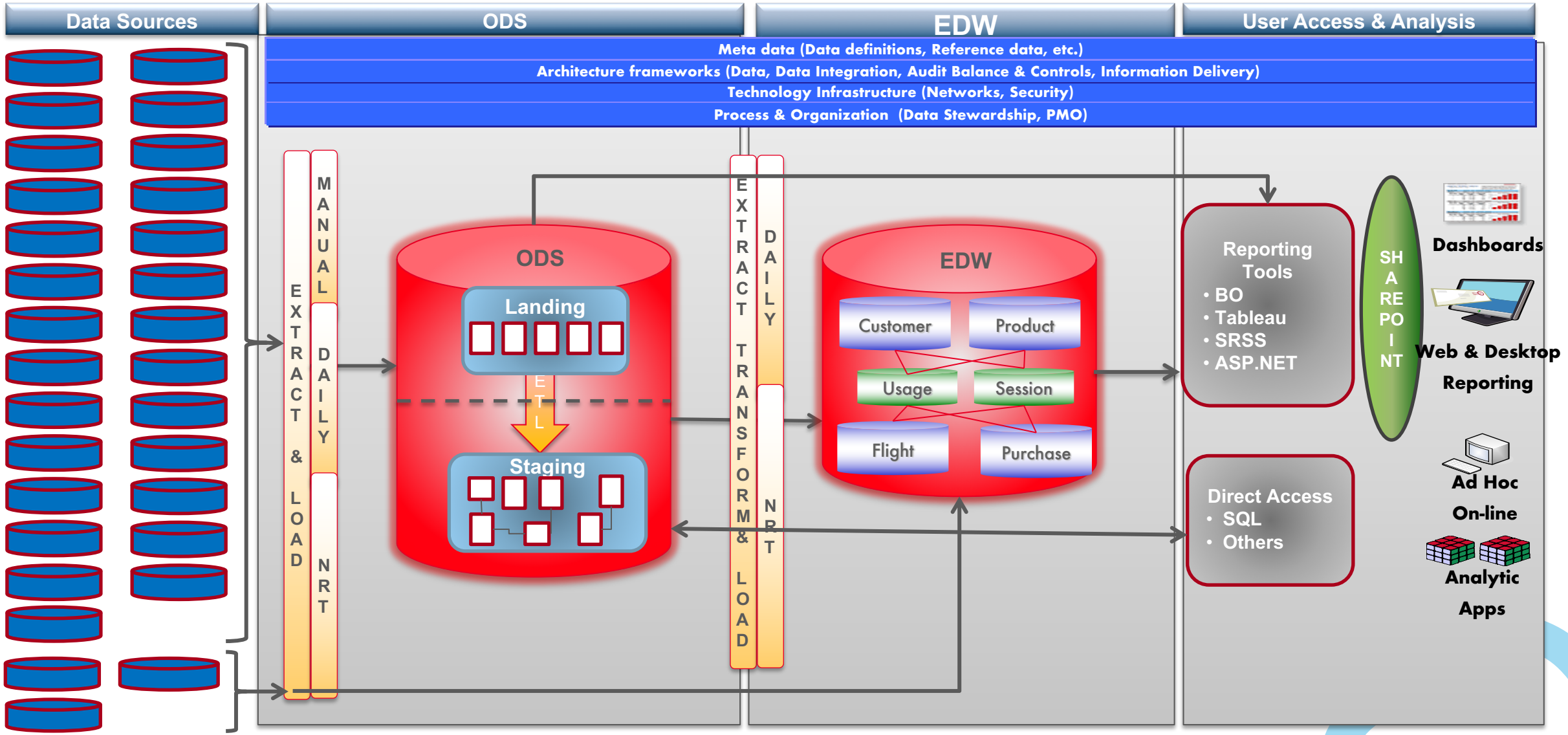
Goals:

- Event-driven data processing
- Near real-time analytics
- Prediction modeling

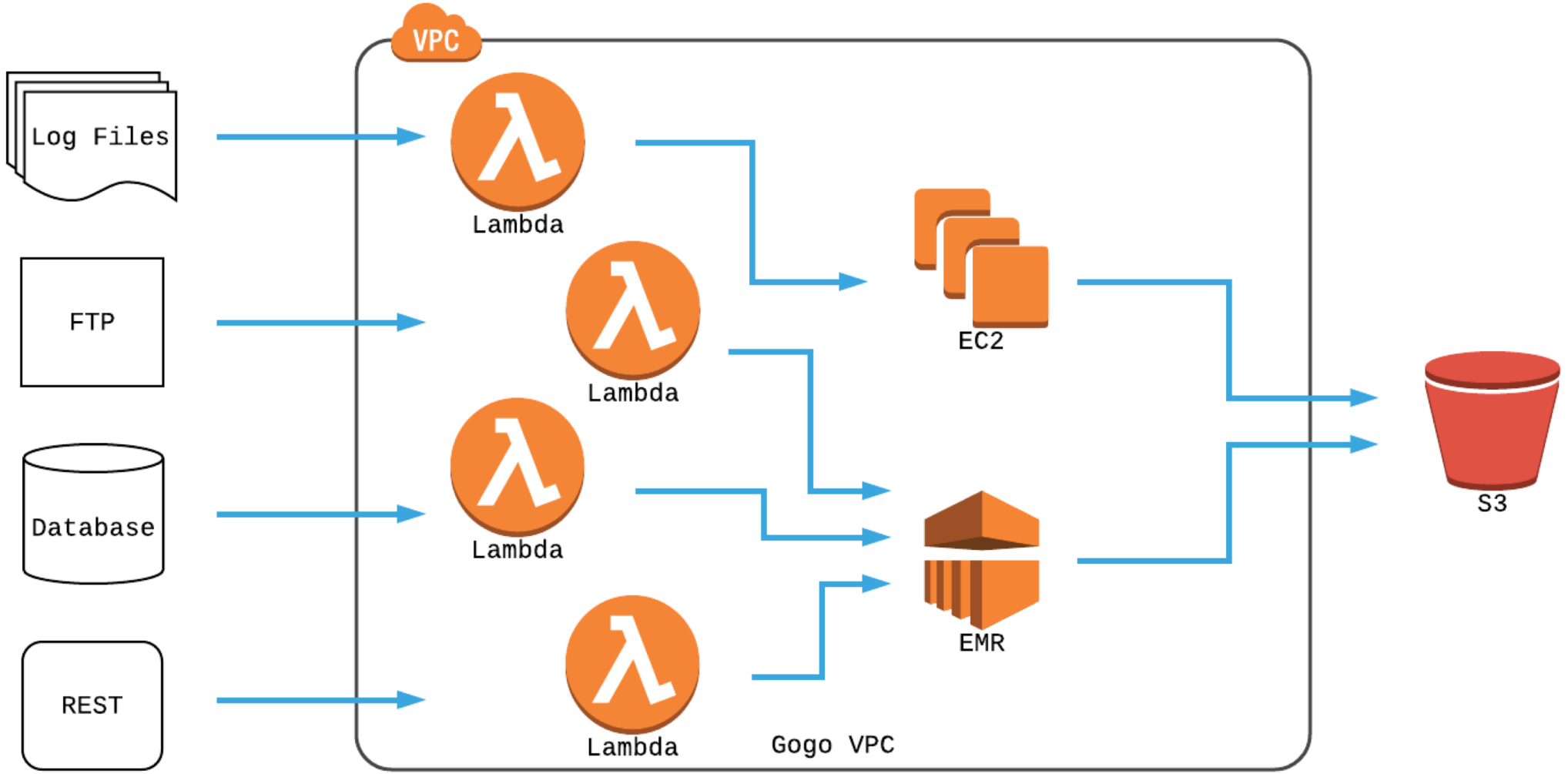
AWS:

- Availability
- Scalability
- Managed Services

Old world

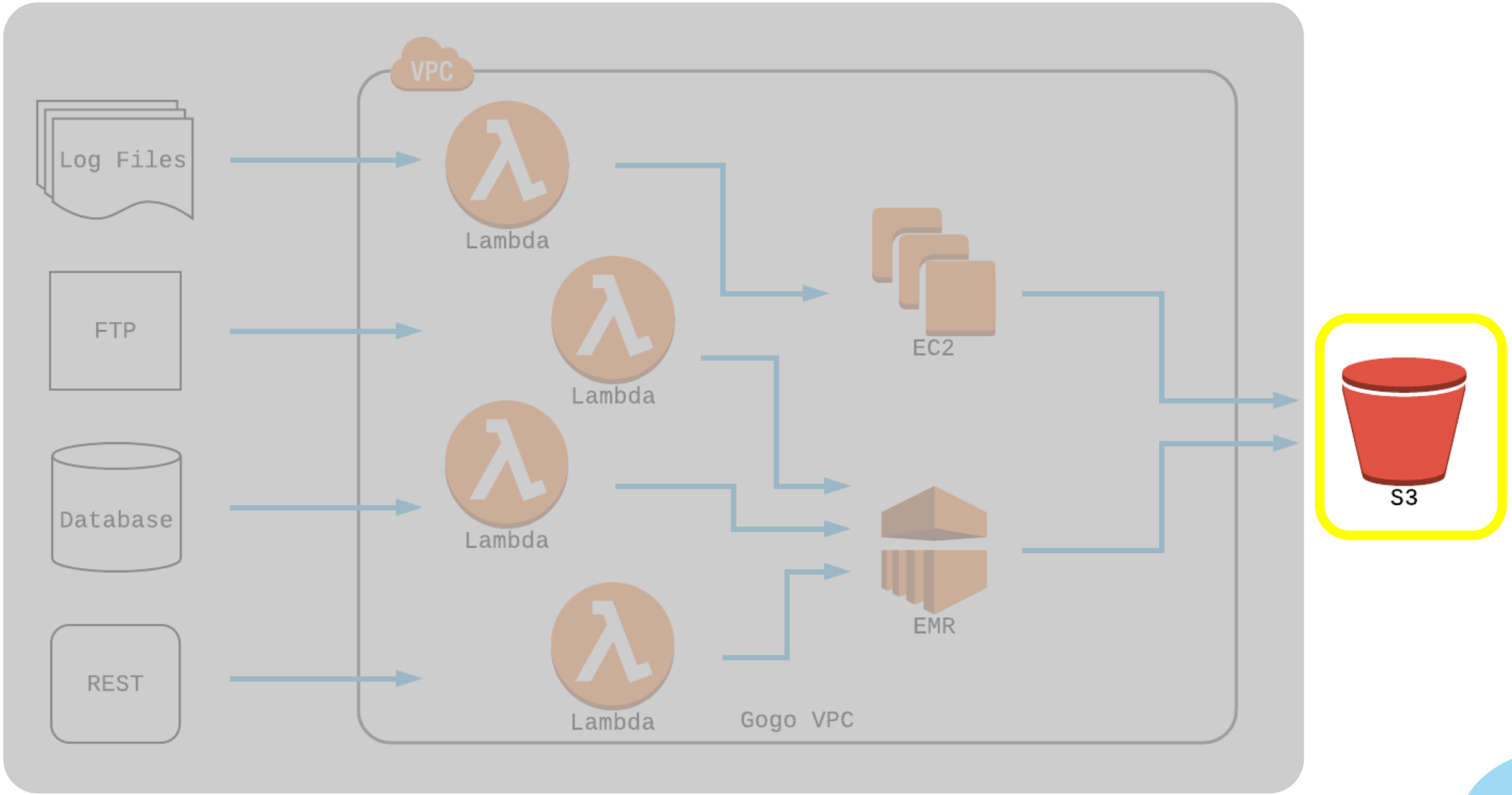


New world



Lessons Learned

S3: The object store



Buckets

- S3 is an object store, not a file system
- Any “directory” structure is nothing more than a string prefix to the object

Issues:

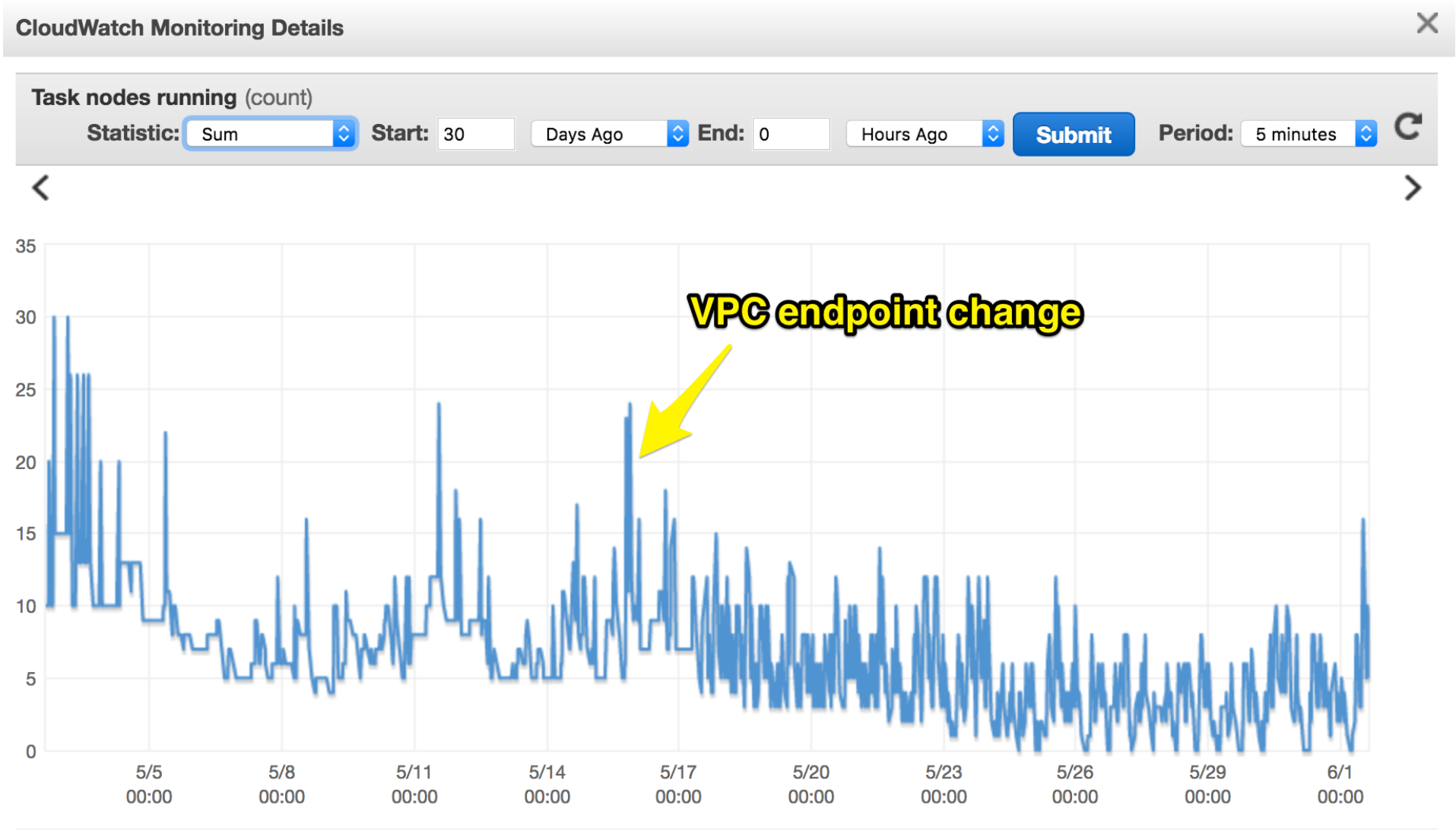
- Bucket contention
- API Rate limiting (500 errors)

Parquet: The joys & the pitfalls

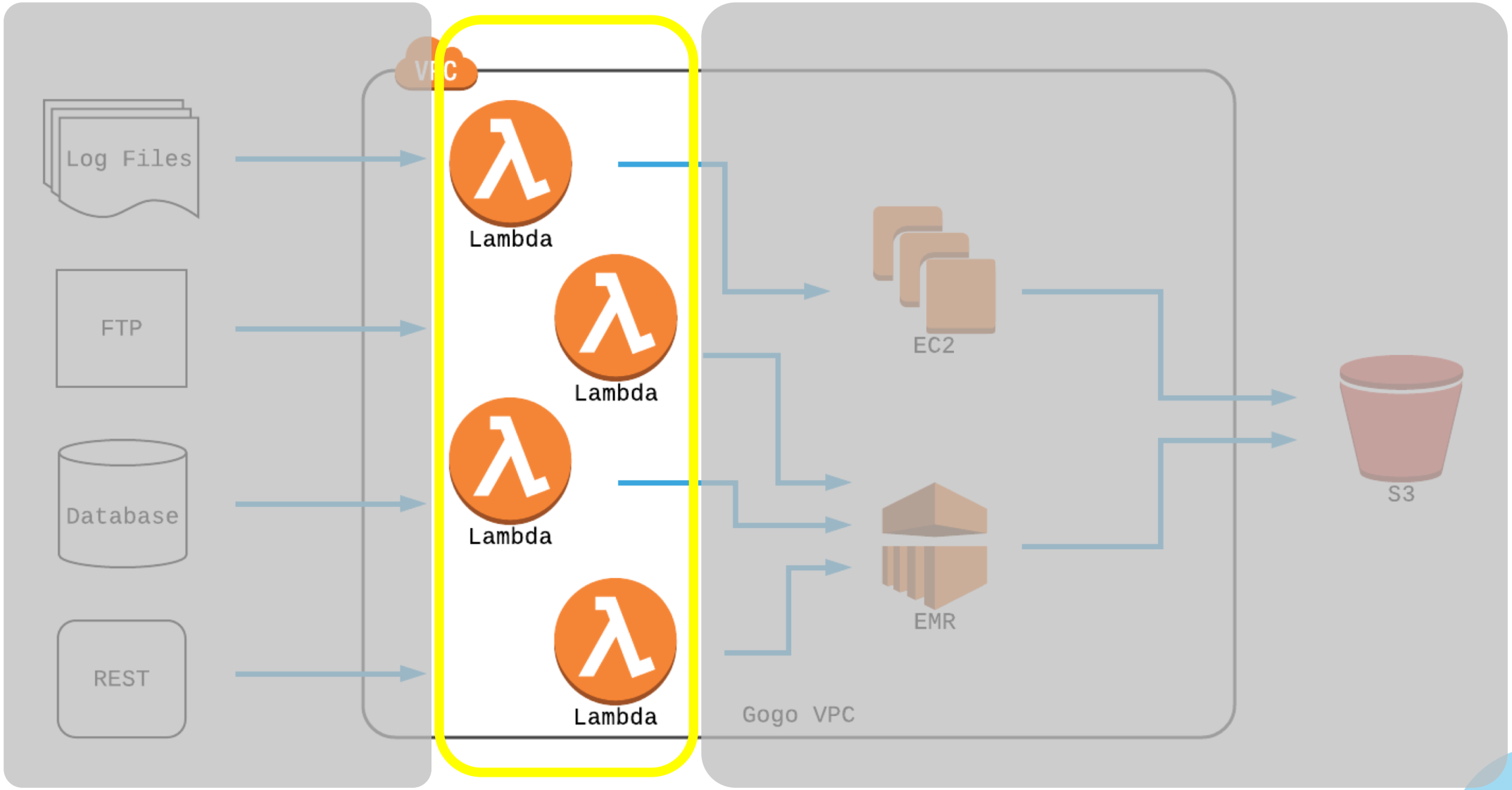
- Compressed, columnar file format
- Natively supported by many analytical tools
  - Pandas
  - Spark
  - Redshift
  - Presto/Drill/Impala

- Cost: High overhead when writing files

VPC Configuration



Lambda: Serverless computing at scale



Lambda: Pros and Cons, Caveats

Pros:

- No need to manage servers
- Integration with key AWS components (S3, SNS)
- Scalable

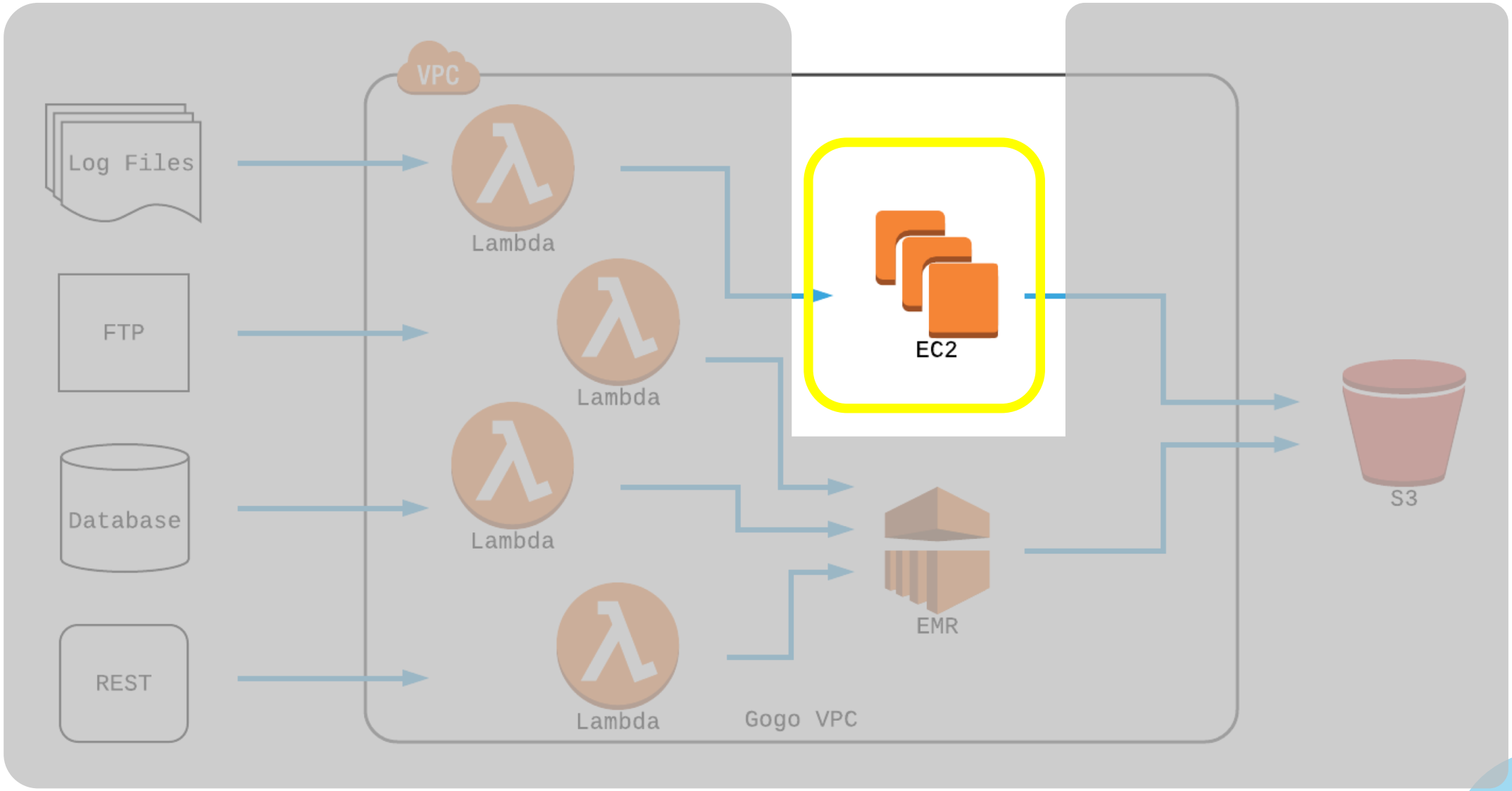
Cons:

- Difficult to test
- Environment is black box
- Scalable

Caveats

- Cost
- Downstream processes (file volumes, API limits)
- Account limits/concurrency

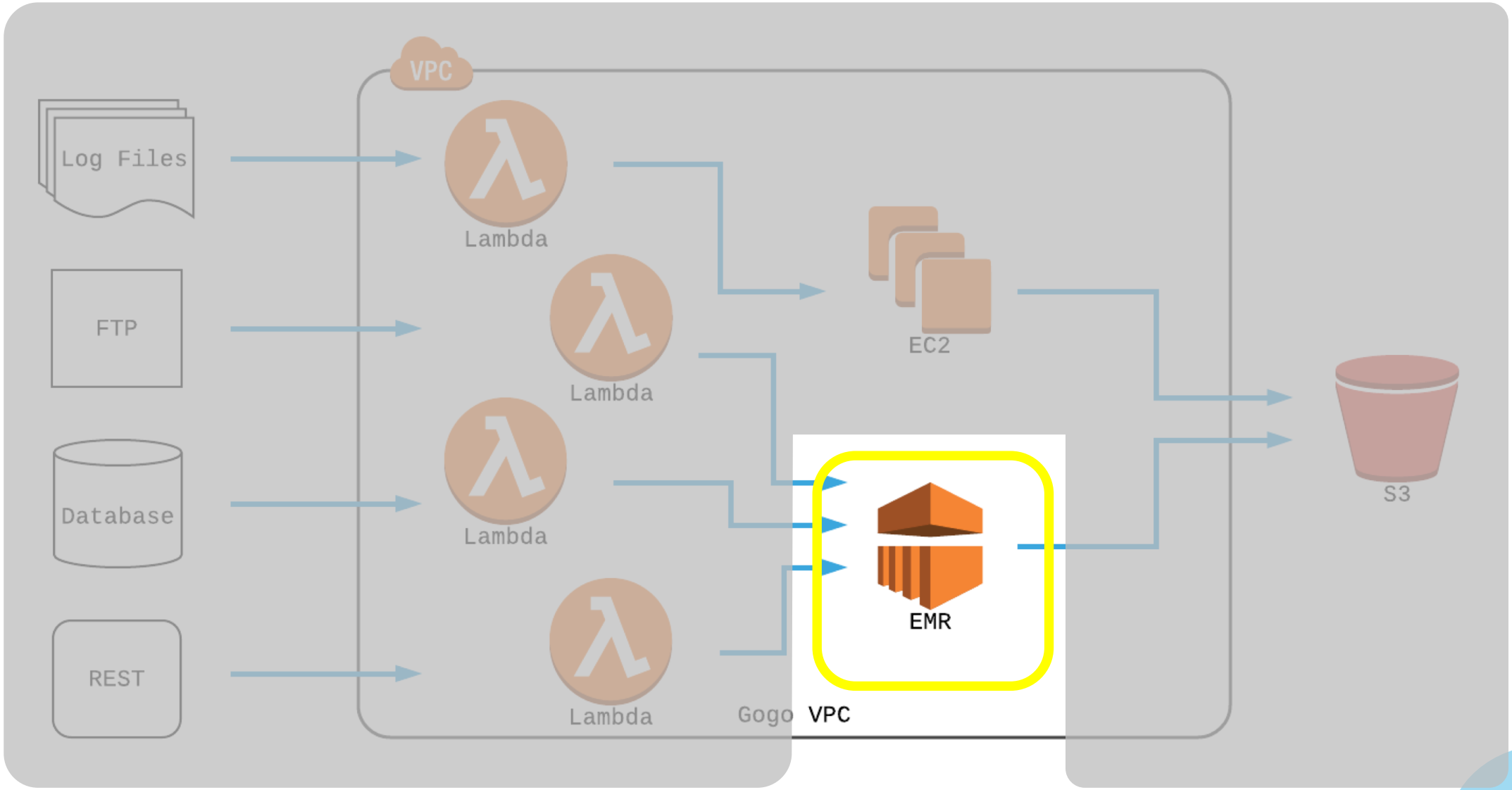
EC2



EC2: Lessons

- Sizing is hard, use more than Cloudwatch to gauge system metrics like memory
- Use spot whenever possible but be aware of the risks
- Spot-pricing can vary dramatically depending on AZ/Region
- Reserving for 1-year increments is a good balance of price/flexibility

EMR: Hadoop as a service



EMR: Lessons

- Step API is limited, geared more toward short-lived clusters – Livy can help
- CI/CD is difficult
- Bigger might be more efficient (because of software costs)
- Scaling is slow/error-prone

Results

- Faster time to reports and analytics
- More ad-hoc analysis
- Similar cost

Other takeaways

- Take small bites
- Set a budget
- Create a local development framework (Docker)

Future State

- Containerization, microservices
- Move away from managed services
- Multi-cloud strategy (GCP, Azure)

Thank you