American Cloud

Cloudstack Conf 2024

Why American Cloud

American Cloud staunchly supports an open and liberated internet. We underscore exceptional service, paramount security, and streamlined processes, letting you focus on achieving the American Dream.

Uncompromised Security
Know your data is safe.

Resilient Backup & Recovery
Know you're one click away from recovery.

Exceptional Support
Dedicated support available around the clock.

Cost Savings
Know that you aren't overpaying Big Tech.

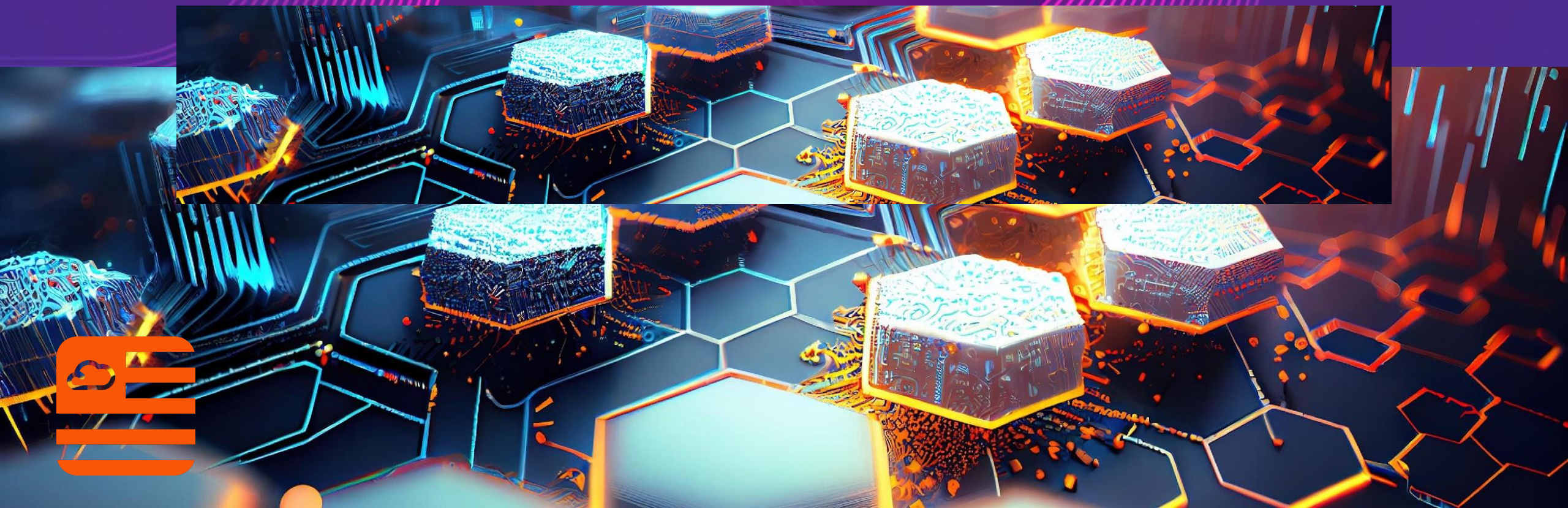
Welcome

Aron Wagner
CEO and Co-Founder

Ben Linzel
Principal Software Engineer

Our mission is to empower businesses of all sizes to achieve their full potential through the use of innovative cloud computing.

Building a Usage Service based off Cloudstack events with Kafka



What did we make?

We built our own real-time, event-driven billing engine based off Cloudstack events

The Problems with Cloudstack Usage Service

Cloudstack has a Usage Service which can be used to generate usage summaries for Cloudstack resources BUT:

Problem #01

It only runs on a schedule (data is delayed for an hour, or a day, depending on config)

Problem #02

It uses up management server resources (cpu, memory, disk)

Problem #03

Doesn't allow for customization of events for billing (do we start billing when VM is created, or starts building? do we stop billing when user clicks delete, or when it's successfully expunged?)

The Problems with Cloudstack Usage Service

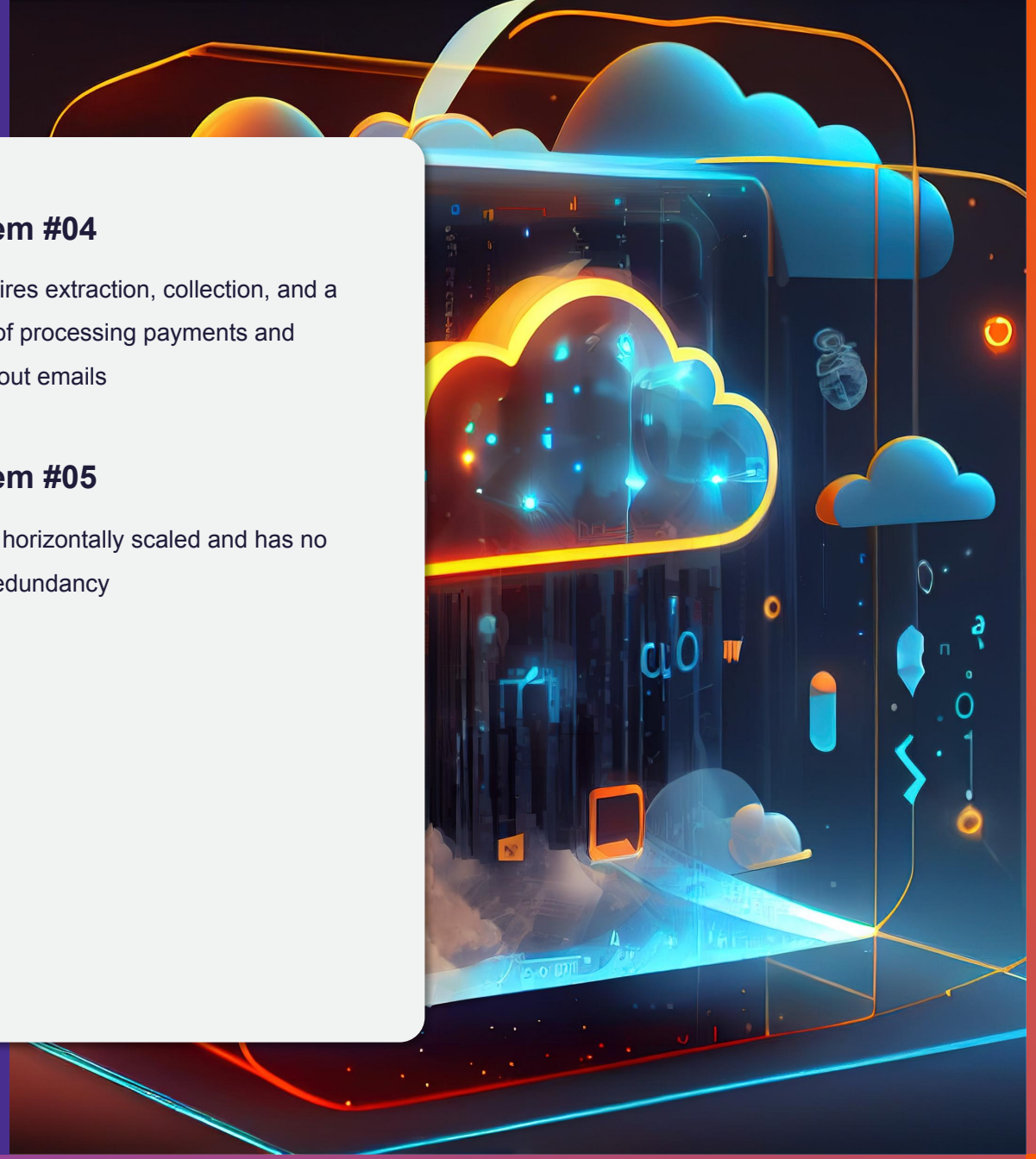
Cloudstack has a Usage Service which can be used to generate usage summaries for Cloudstack resources BUT:

Problem #04

Still requires extraction, collection, and a method of processing payments and sending out emails

Problem #05

Can’t be horizontally scaled and has no built-in redundancy



The Solution

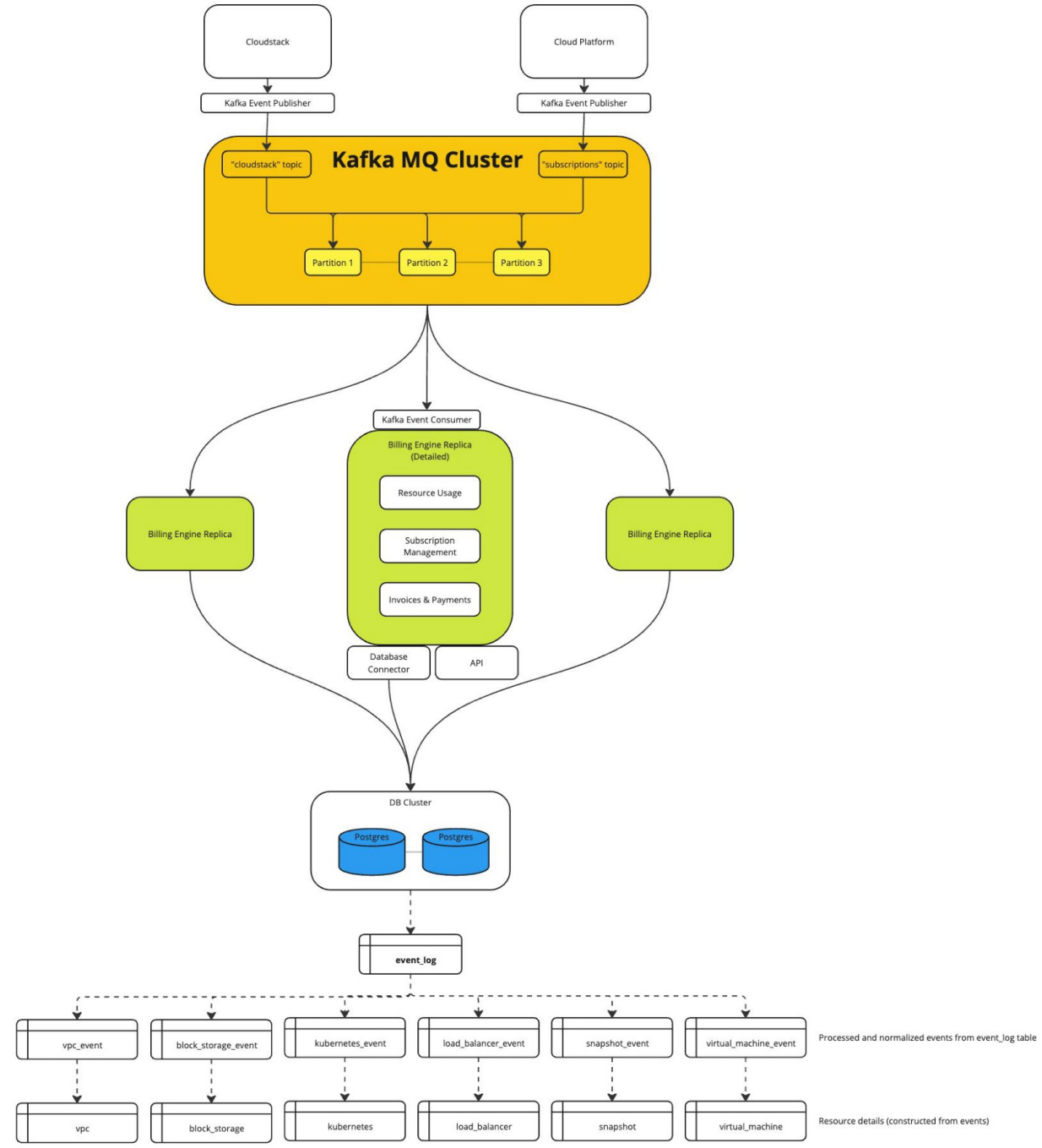
We inspected the schema of Cloudstack events, and realized for every resource (VMs, VPCs, Volumes, Networks, Kubernetes) the events gave us sufficient information to drive real-time metering and billing

We decided to subscribe to these events using the Cloudstack Event Bus

We picked Kafka over RabbitMQ for its maturity and strong built-in scaling, HA, and replay capabilities, but we wrote a Rabbit adapter as well.

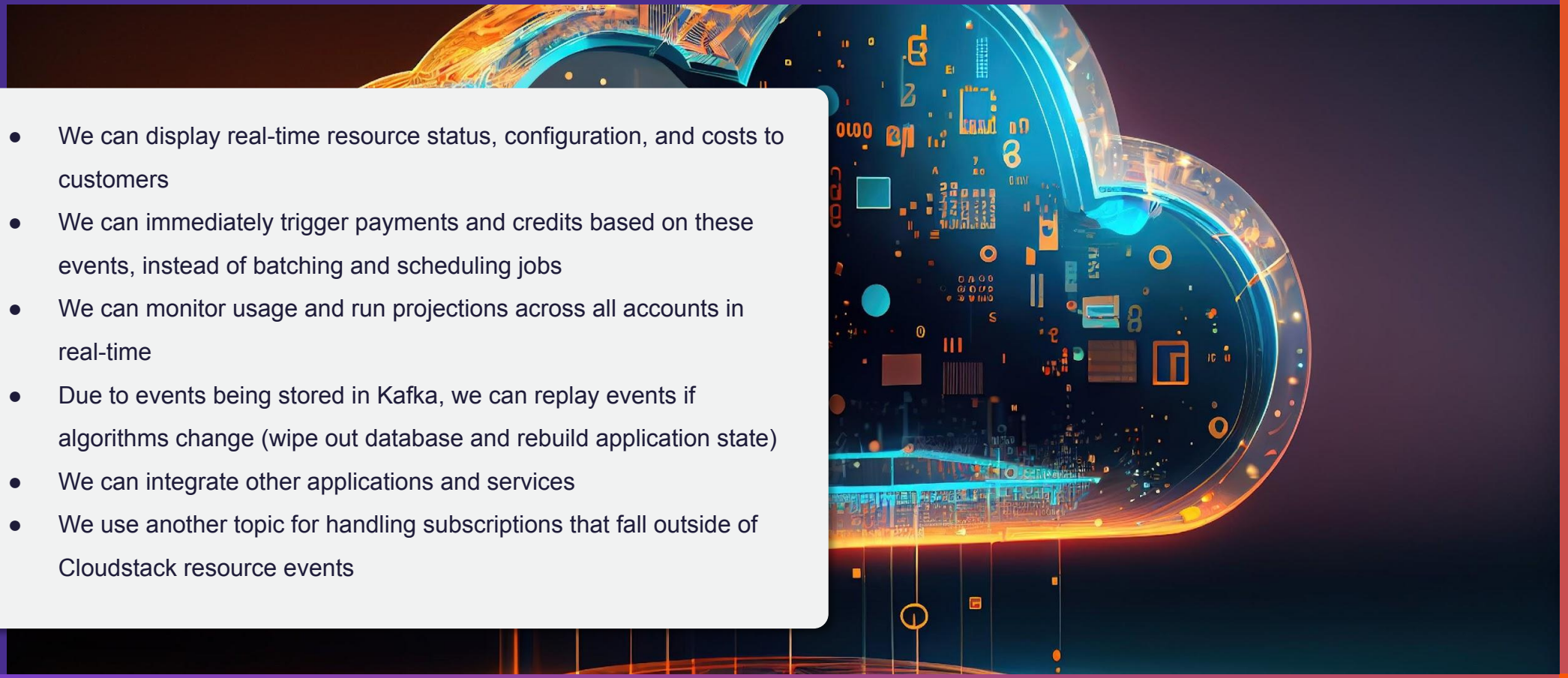
The Pub/Sub model also allows us to scale our engine, since events will only ever be delivered to a single replica

The replicas point to what they see as a single database, and process the events and update the state of the resource in the event

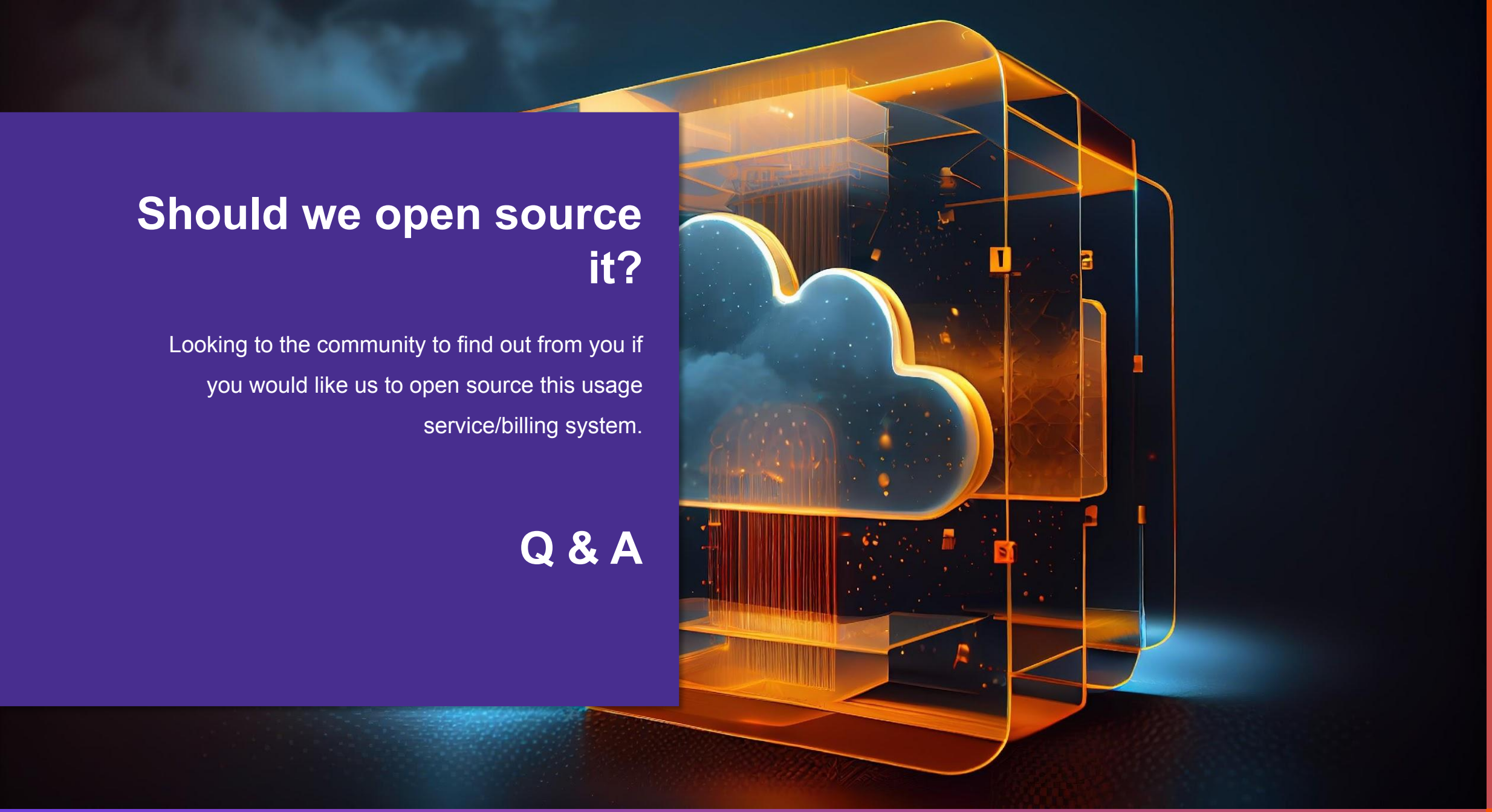

Cloudstack
Cloud Platform
Kafka Event Publisher
Kafka Event Publisher
"cloudstack" topic
Kafka MQ Cluster
"subscriptions" topic
Partition 1
Partition 2
Partition 3
Kafka Event Consumer
Billing Engine Replica (Detailed)
Resource Usage
Subscription Management
Invoices & Payments
Billing Engine Replica
Billing Engine Replica
Database Connector
API
DB Cluster
Postgres
Postgres
event_log
vpc_event
block_storage_event
kubernetes_event
load_balancer_event
snapshot_event
virtual_machine_event
Processed and normalized events from event_log table
vpc
block_storage
kubernetes
load_balancer
snapshot
virtual_machine
Resource details (constructed from events)

What can we do with it?

"Whether you think you can or think you can't. you're right"



- We can display real-time resource status, configuration, and costs to customers
- We can immediately trigger payments and credits based on these events, instead of batching and scheduling jobs
- We can monitor usage and run projections across all accounts in real-time
- Due to events being stored in Kafka, we can replay events if algorithms change (wipe out database and rebuild application state)
- We can integrate other applications and services
- We use another topic for handling subscriptions that fall outside of Cloudstack resource events



Should we open source it?

Looking to the community to find out from you if you would like us to open source this usage service/billing system.

Q & A

Contact

Aron Wagner
CEO & Co-Founder
aron@americancloud.io

THANK YOU

American Cloud