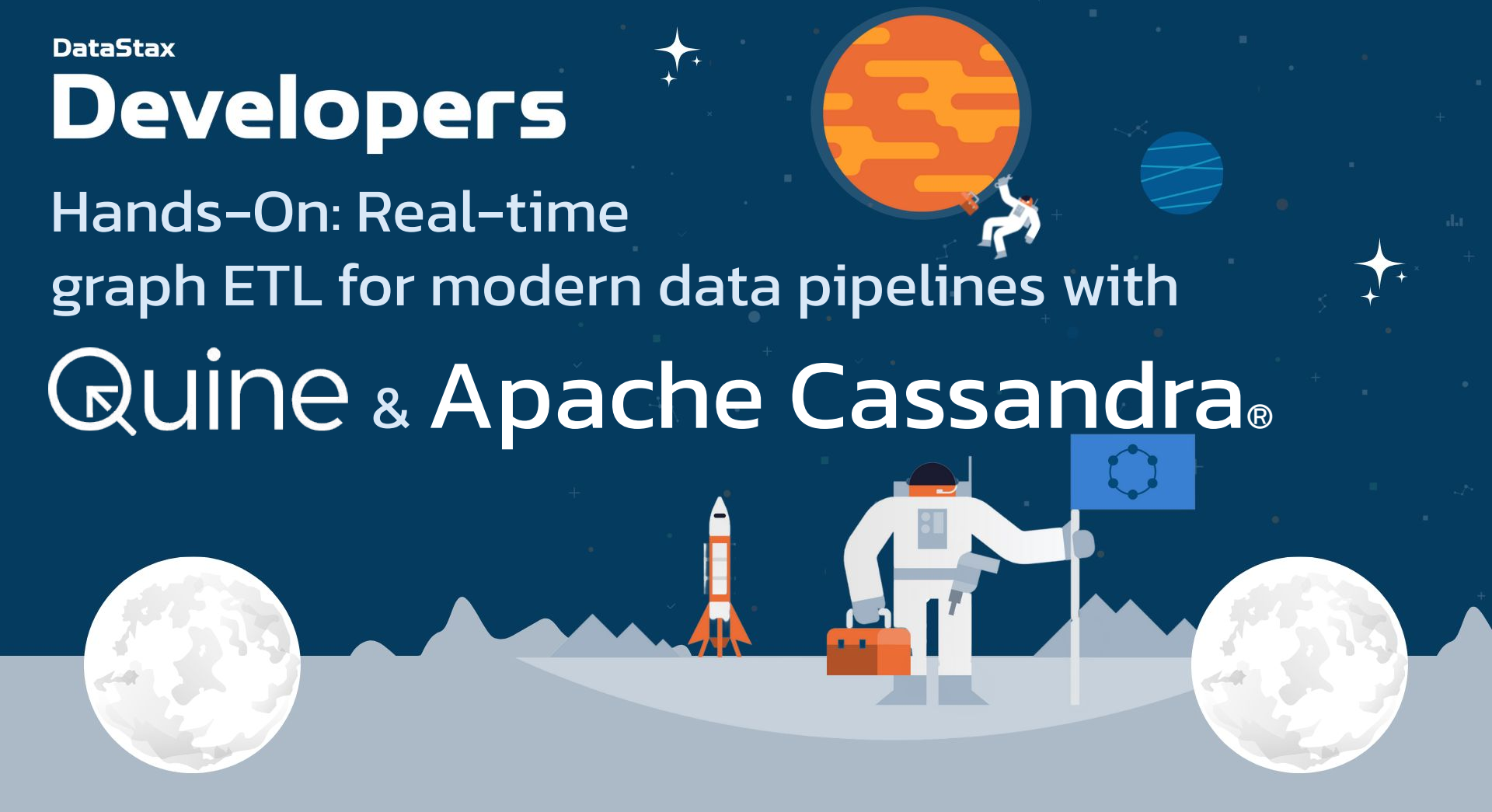
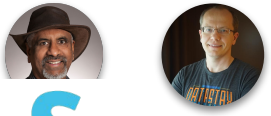
DataStax

Developers

Hands-On: Real-time graph ETL for modern data pipelines with Quine & Apache Cassandra®





S S T S

David Dieruf

Rags Srinivas

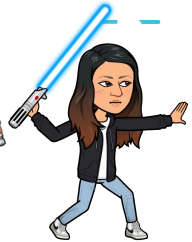
Artem Chebotko

Stefano Lottini

Aleksandr Volochnev

Aaron Ploetz

S S S



Jack Fryer

Kirsten Hunter

Cedrick Lunven

Mary Grygleski

Ryan Welford

David Gilardi

DataStax Developers Crew

DataStax Developers

Developer Advocate

DataStax

cassandra

@aaronploetz

@aploetz

@aploetz

- Former SWE/DevOps/DB Lead @ GRAINGER & TARGET
- Host – Apache Cassandra Corner podcast
- Worked as an author on:
  - Mastering Apache Cassandra 3.x
  - Seven NoSQL Databases in a Week

Aaron Ploetz

Developer Relations

thatDot

cassandra

Quine

@michael-aglietti

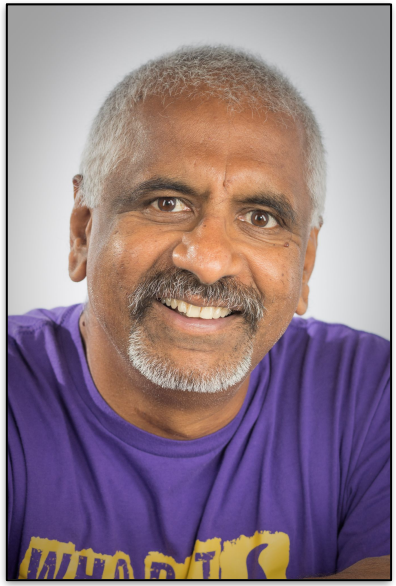
@maglietti

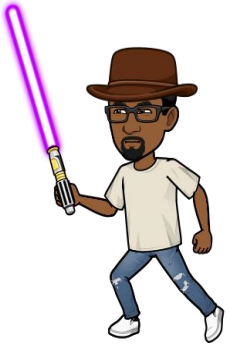
@michaelaglietti

- DevRel Leader
- Data-driven fanatic
- Solutions Architect
- Systems Engineer
- Information Architecture
- LAMP stack developer

Michael Aglietti

Developer Advocate





- Developer/Architect
- Mechanical Engineer (so many moons ago)
- Distributed systems
- Love to teach and communicate
- Inner loop == developer productivity

@rags

@ragsns

@ragss

Raghavan “Rags” Srinivas

And a badge for your pleasure!

01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

Hands on

06

What's next?

Quiz, Homework, Next week

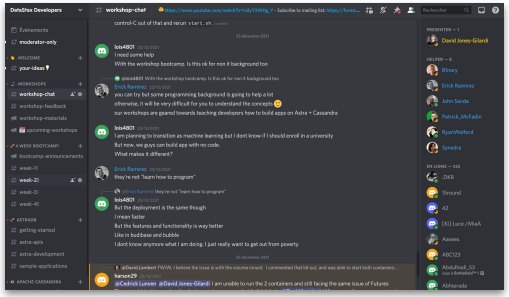
Agenda

Livestream: youtube.com/DataStaxDevs

YouTube

Twitch

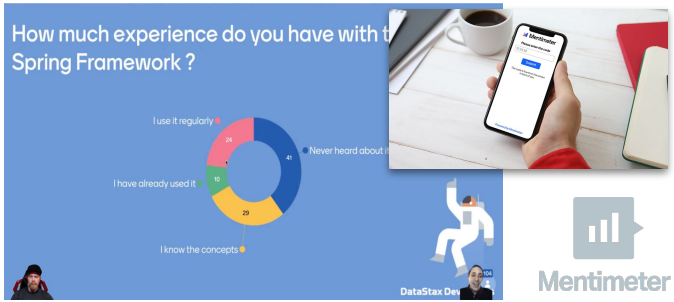
Questions: https://dtsx.io/discord



YouTube

Discord

Games and quizzes: menti.com



Mentimeter

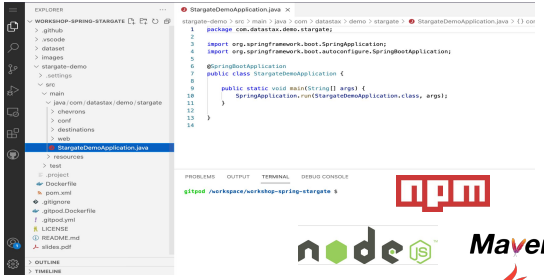
1 Attend the live sessions

AstraDB astra.datastax.com

Database + GraphQL + PlayGround

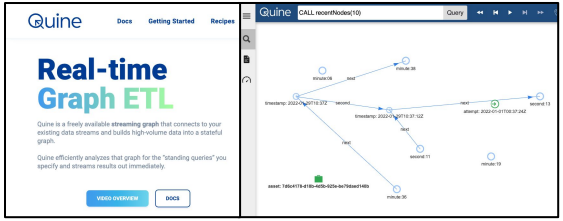
DataStax
Astra DB

Gitpod IDE: https://gitpod.io

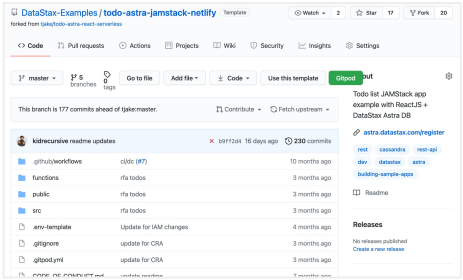


npm
node
Maven
Gitpod

Quine https://quine.io/



Quine



GitHub

2 Complete Workshops Labs

Today's Workshop: Quine & Cassandra

01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

Hands on

06

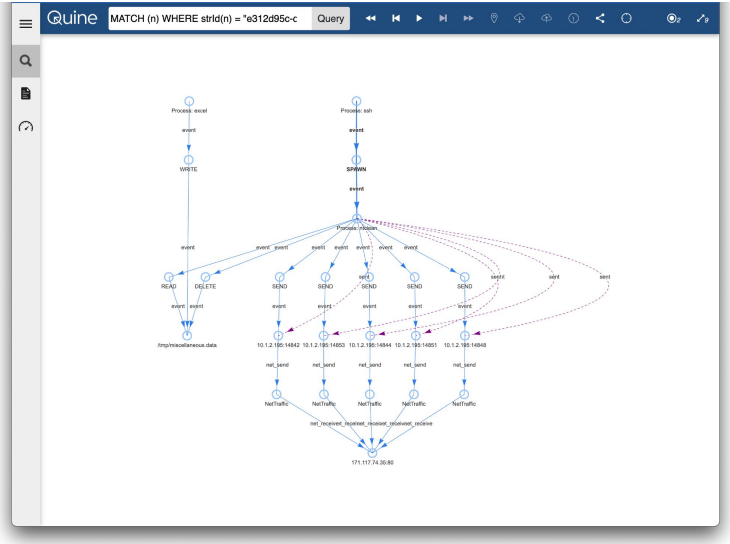
What's next?

Quiz, Homework, Next week

Agenda

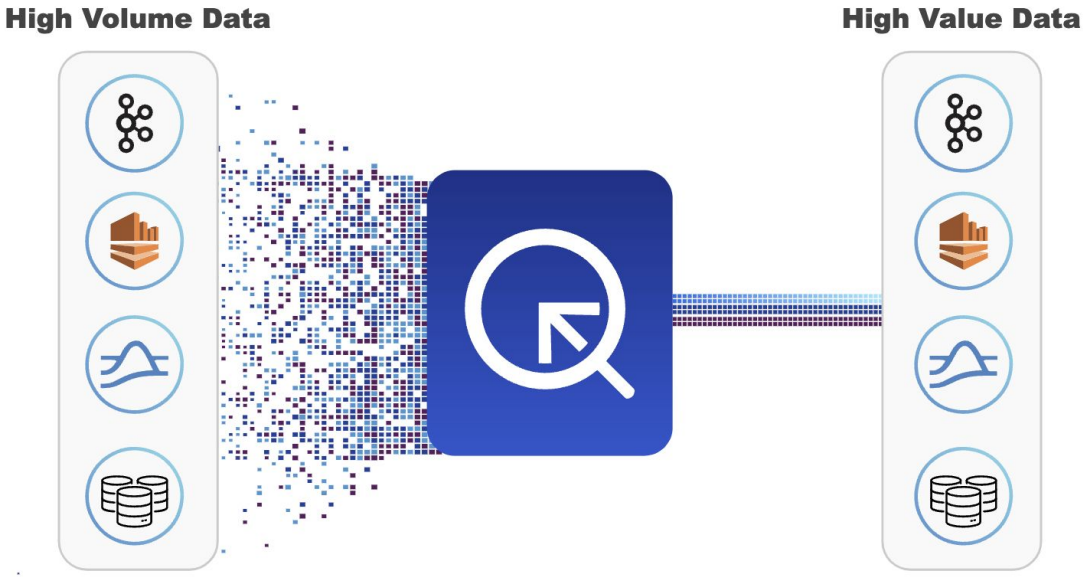
Quine

- Open-source Graph ETL platform
- Graph data model easily joins multiple data sets
  - Track complex categorical relationships
  - No time windows to manage
  - Automated out-of-order data handling
  - High cardinality data sets
  - Flexible data structure
- Standing queries operate on data in real-time, or to generate events



What is Quine?

Architecturally, Quine fits into a streaming event pipeline in between streaming event processors



Quine can shape, filter, analyze, or take action based on patterns matched in the event stream

01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

Hands on

06

What's next?

Quiz, Homework, Next week

Agenda

Log Analysis
General IT

Combine logs from multiple services to understand system performance issues and their root causes.

Streaming ETL
Event Streaming Architecture

Real-time ETL on streaming data, when data is received from multiple sources, requires new tools to accommodate out-of-order data.

Video Observability
Online Video

Video delivery monitoring is hard. Logs come from multiple sources with high dimensionality and cardinality, at scale.

Fraud Detection
Finance

Financial fraud detection typically relies on analysis of event volumes and transactions bypassing much of the categorical data.

Auth Fraud
General IT

Low frequency authentication attacks, such as password spraying and credential stuffing, circumvent authentication policies.

Blockchain Fraud
Finance - Blockchain

Monitor, detect and trace fraudulent transactions across blockchains in real time.

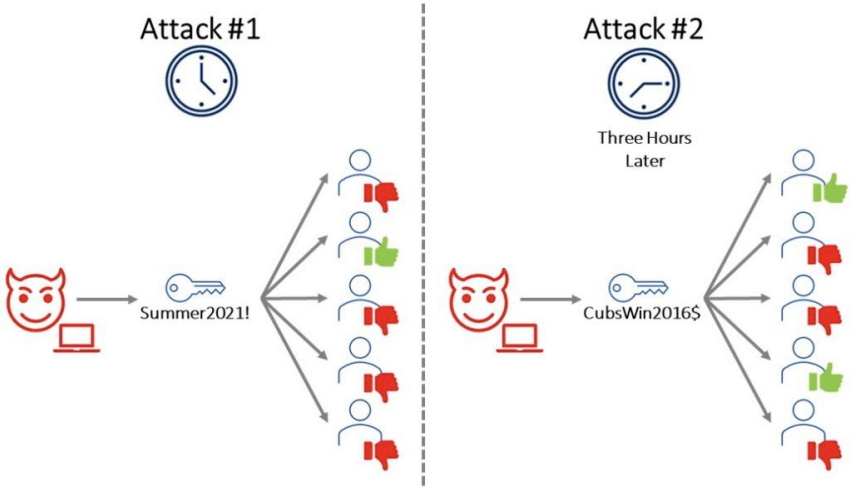
Quine

Value for Data Driven Businesses

Password Spraying

According to Microsoft, the three steps to conduct a password-spraying attack are:

- **Acquire a list of usernames**: starting with a list of names: firstname.lastname@company.com
- **Spray passwords**: testing popular and common passwords (e.g., abc123, password, and Summer21!). See the top 10,000 passwords.)
- **Gain access**: one of the tested attempts works, and the account can be abused to enumerate assets in the AD network, exploit authenticated services and put the organization at risk.



01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

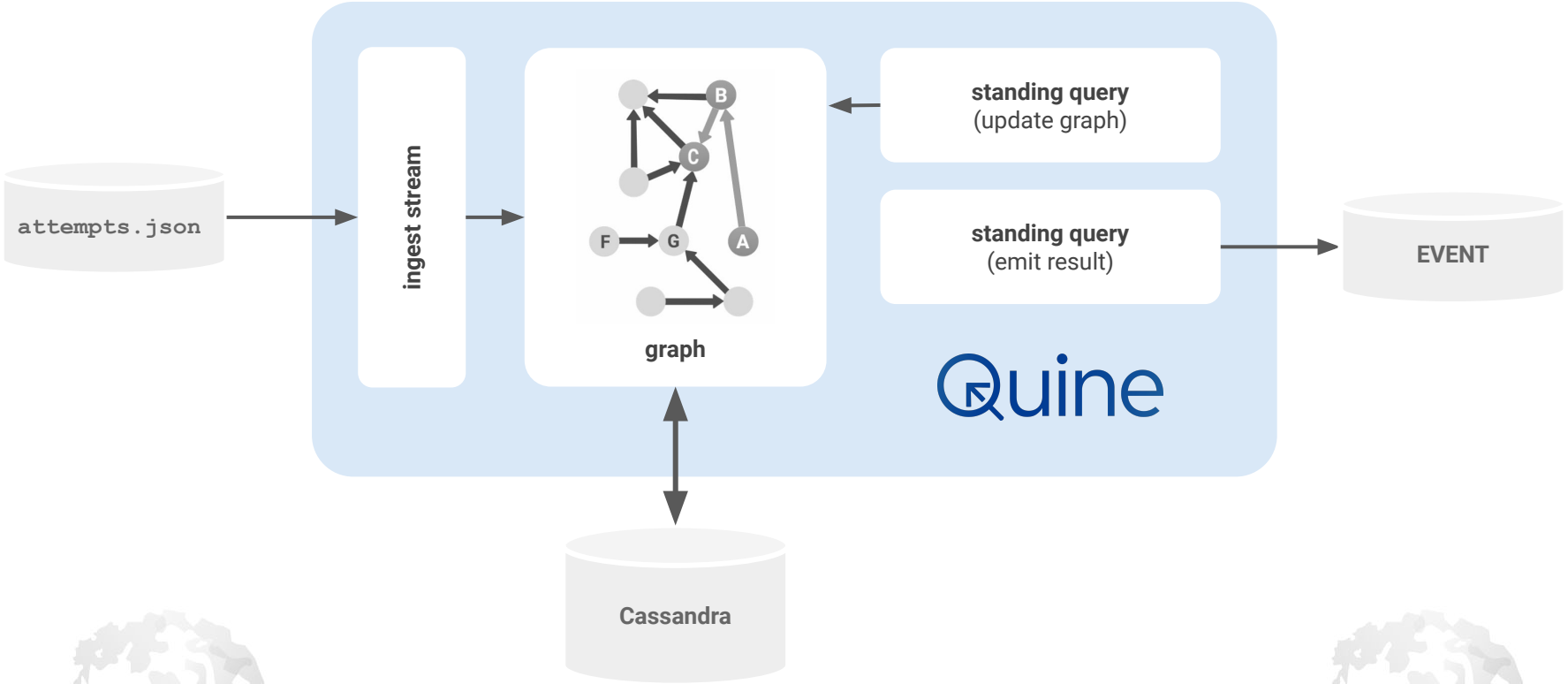
Hands on

06

What's next?

Quiz, Homework, Next week

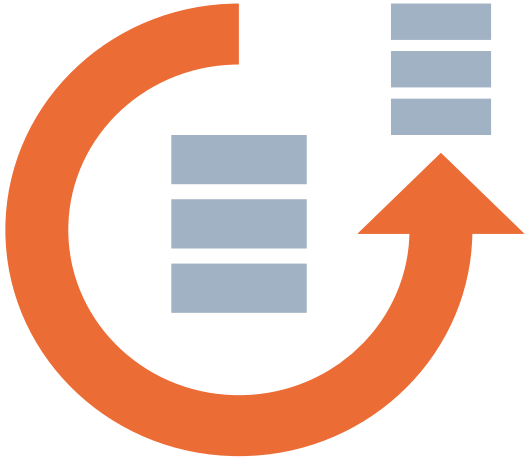
Agenda



Architecture

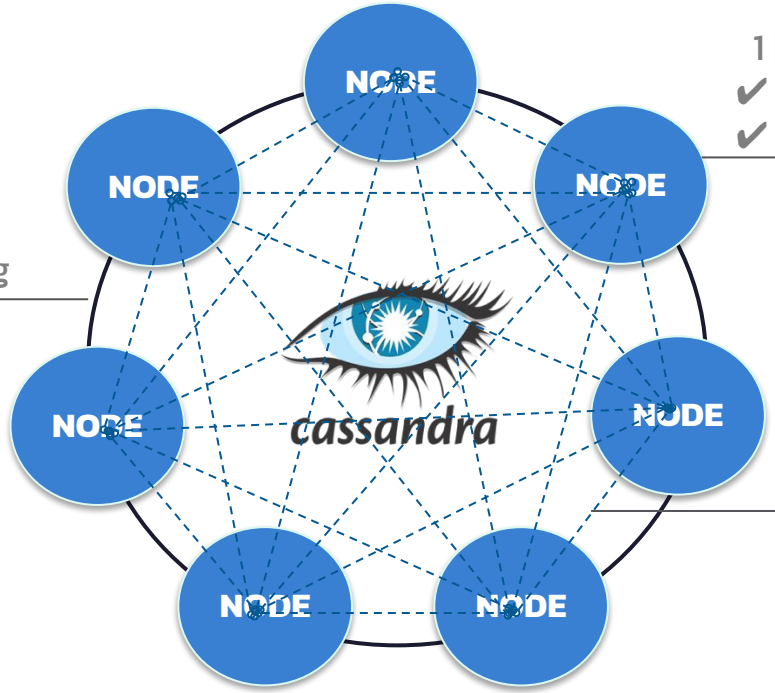
Persistence Layer Options:

- RocksDB
- MapDB
- Apache Cassandra®



Quine storage options

Apache Cassandra®



1 Installation = 1 NODE
- ✔ Capacity = ~ 2TB
- ✔ Throughput = LOTS Tx/sec/core

DataCenter | Ring

Communication:
- ✔ Gossiping

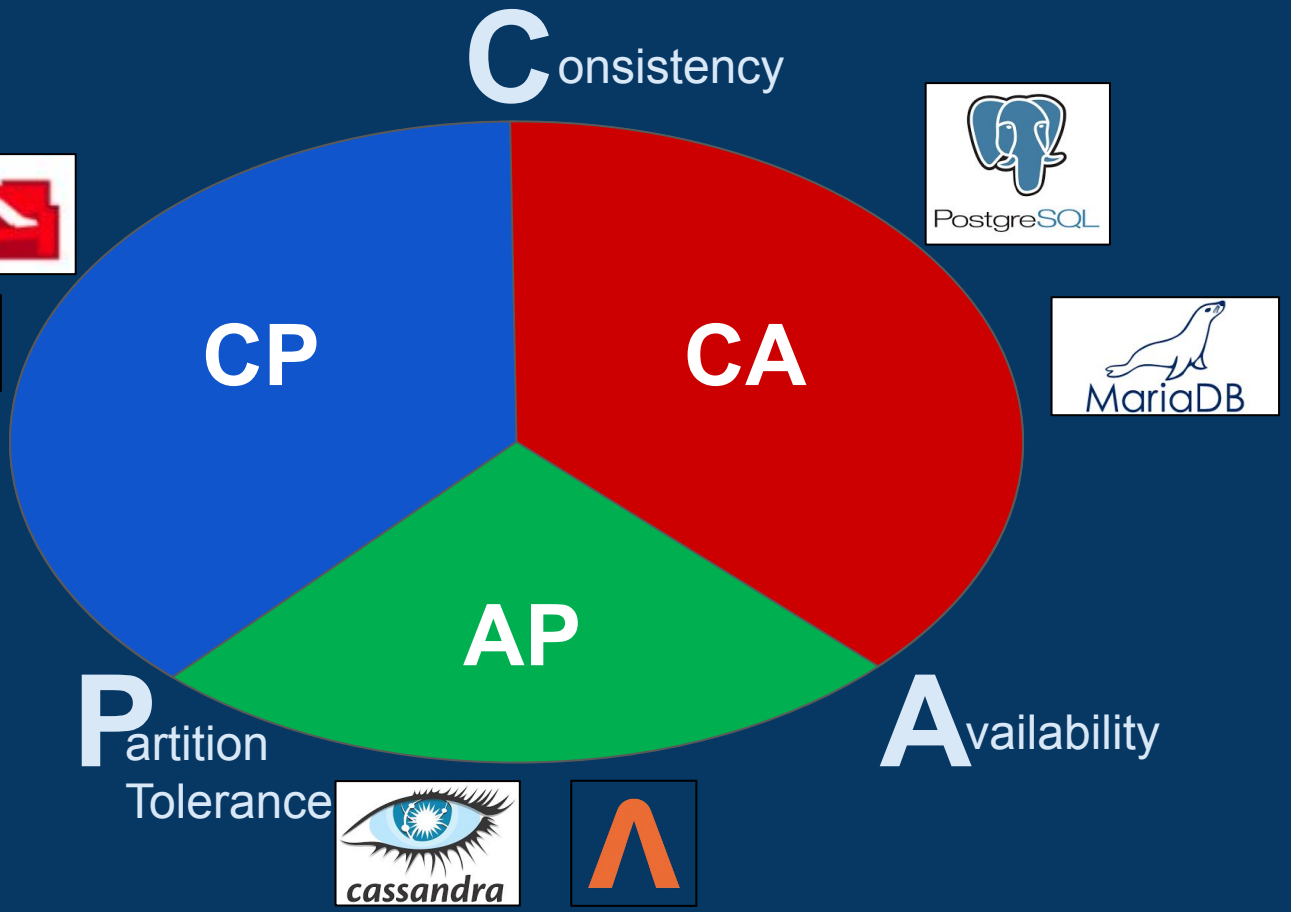
Database: Cassandra

Astra DB:

- Cassandra as a Service!
- No operational overhead
- Geographic Awareness
- Not cloud vendor bound
- “No touch” Scalability

Brewer's CAP Theorem

All distributed systems try to achieve consistency, availability and partition tolerance. But realistically they can only pick two.

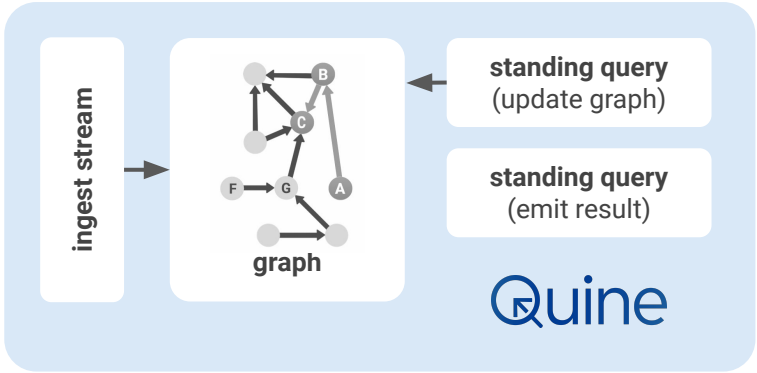


Harvest, Yield, and Scalable Tolerant Systems (1999)

CAP Theorem and Astra DB

Event Streams and Indexes

- Quine shapes event streams – data that is **continuously** generated, often in high **volumes** and at high **velocity** – into a graph.
- Traditional node indexing methods are ***slow***.
- That is why we use a custom Cypher function idFrom which implements a consistent hash to **deterministically** locate a node ID
- Eliminates querying for node IDs
- Cornerstone concept for Quine Cypher queries

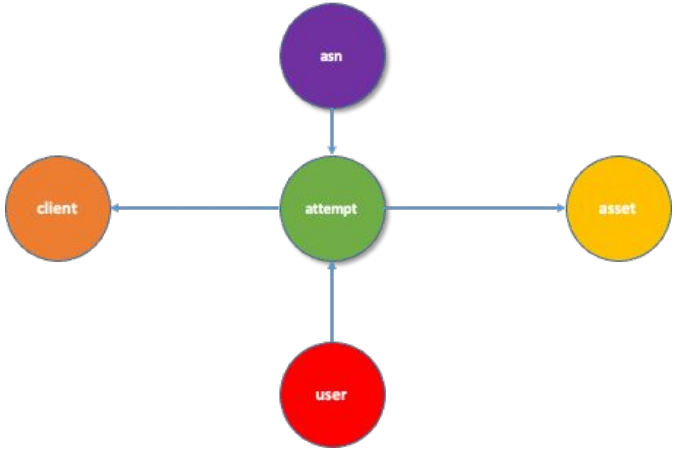



```Cypher
MATCH (n) WHERE id(n) = idFrom($that) SET n.line = $that
```

Key Concept – idFrom()

Attempt Event


```
{
  "schemaVersion": 1,
  "eventId": "c7bfd702-88e3-47a5-93c5-8b699e803674",
  "timestamp": "2022-01-01T00:08:11Z",
  "level": "info",
  "eventCode": 1,
  "eventType": "user.session.auth",
  "displayMessage": "User login",
  "transaction": {
    "type": "LDAP",
    "entityId": "fd5296ee-346a-41be-b2a8-10f10319cd45",
    "id": "760f08c0-d458-4a31-97ff-5856a1b269e6"
  },
  "eventTypeDetail": "user.authentication.login.success",
  "outcome": {
    "result": "SUCCESS",
    "details": ""
  },
  "client": {
    "zone": "VPN",
    "asn": 12414,
    "ipAddress": "220.209.114.209",
    "requestUri": "/rhoncus.js?leo=faucibus&maecenas=orci&pulvinar=luctus&lobortis=et&est=ultrices&phasellus=posuere&sit=cubilia&amet=curae&erat=mauris&nulla=viverra&tempus=diam&vivamus=vitae&in=quam&felis=suspendisse&eu=potenti&sapien=nullam&cursus=porttitor&vestibulum=lacus&proin=at&eu=turpis&mi=donec&nulla=posuere&ac=metus&enim=vitae&in=ipsum&tempor=aliquam",
    "device": "unknown",
    "userAgent": "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.792.0 Safari/535.1"
  },
  "user": {
    "id": "9c3547ae-e363-4fe6-893f-87caae67d186",
    "type": "Admin",
    "alternateId": "Caro.Howell@acme.com",
    "displayName": "Caro Howell"
  }
}
```

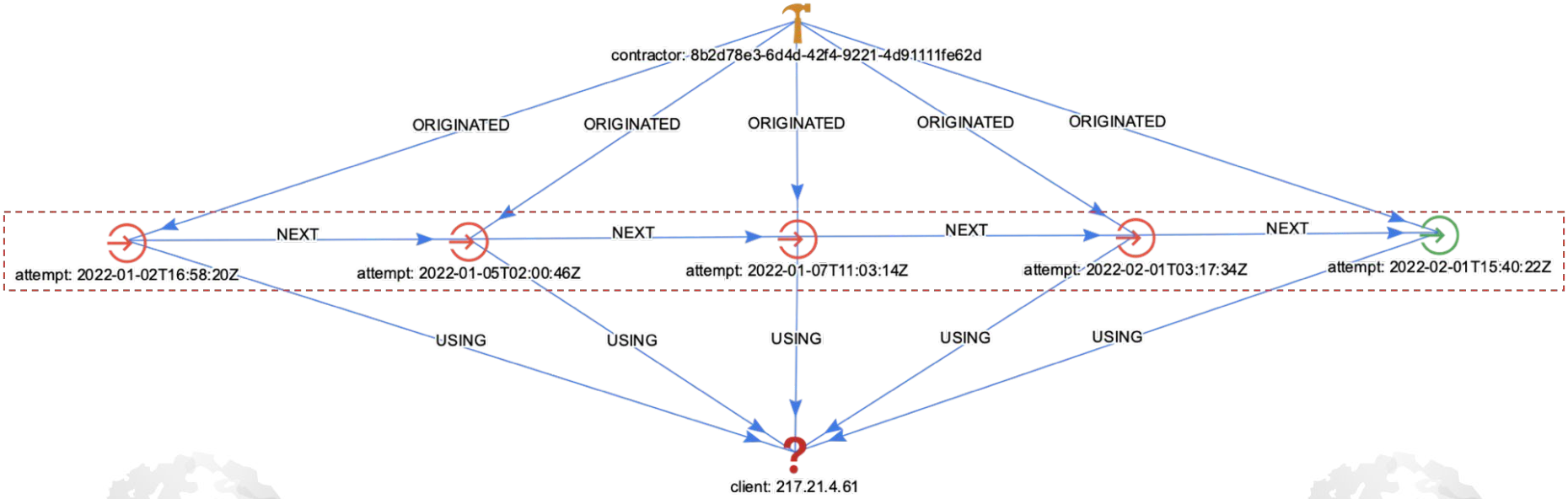



The **attempt** event is the most atomic unit. It represents a single log entry.

All nodes are materialized from the **attempt** data during ingest using Cypher to create nodes, properties, and relationships.

Event -> Ingest Stream

Four sequential **failed** attempts from the same user/client pair



```
- pattern:
    type: Cypher
    query: |-
      ////////////////////////////////////////////////////////
      // Subquery to match 4 consecutive failed attempts followed by a successful attempt
      ////////////////////////////////////////////////////////
      MATCH (attempt1 {outcomeResult: "FAILURE"})-[:NEXT]->(attempt2 {outcomeResult: "FAILURE"})-[:NEXT]->(attempt3 {outcomeResult:
      "FAILURE"})-[:NEXT]->(attempt4 {outcomeResult: "FAILURE"})-[:NEXT]->(attempt5 {outcomeResult: "SUCCESS"})
      RETURN DISTINCT id(attempt1) AS attempt1
    mode: DistinctId
  outputs:
    alert:
      type: CypherQuery
      query: |-
        MATCH (attempt1 {outcomeResult:"FAILURE"})-[:NEXT]->(attempt2 {outcomeResult:"FAILURE"})-[:NEXT]->(attempt3
        {outcomeResult:"FAILURE"})-[:NEXT]->(attempt4 {outcomeResult:"FAILURE"})-[:NEXT]->(attempt5 {outcomeResult:"SUCCESS"})
        WHERE id(attempt1)=$that.data.attempt1
        RETURN 'Password Spraying Attack: ' + 'http://localhost:8080/#MATCH%20(user)-[:ORIGINATED]-%3E
        (attempt1%20%7BoutcomeResult:%22FAILURE%22%7D)-[:NEXT]-%3E(attempt2%20%7BoutcomeResult:%22FAILURE%22%7D)-[:NEXT]-%3E
        (attempt3%20%7BoutcomeResult:%22FAILURE%22%7D)-[:NEXT]-%3E(attempt4%20%7BoutcomeResult:%22FAILURE%22%7D)-[:NEXT]-%3E
        (attempt5%20%7BoutcomeResult:%22SUCCESS%22%7D)%20WHERE%20id(attempt1)=%22' + toString(strId(attempt1)) +
        '%22%20RETURN%20DISTINCT%20user%2Cattempt1%2Cattempt2%2Cattempt3%2Cattempt4%2Cattempt5' AS QuineUILink
      andThen:
        type: PrintToStandardOut
```

The Bad Pattern

01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

Hands on

06

What's next?

Quiz, Homework, Next week

Agenda

Astra DB

Jump to README steps #4 and #5

https://github.com/datastaxdevs/workshop-streaming-graph-quine#create-your-astra-db-instance

https://github.com/datastaxdevs/workshop-streaming-graph-quine#setup-quine

Hands On


workshop-streaming-graph-quine
EXPLORER
Preview README.md
start.sh
OPEN EDITORS
Preview README.md
start.sh
WORKSHOP-ST...
.vscode
data
.gitignore
.gitpod.Dockerfile
.gitpod.yml
attempts.json
get_quine.sh
password-spraying-worksho...
quine-1.3.2.jar
quine.conf
quine.jar
README.md
start.sh
stats.sh
stop.sh
OUTLINE
TIMELINE
quine-content-dev*
Spell
Graph Streaming with Astra and Quine
Welcome to the real-time graph ETL for modern data pipelines with Quine and Apache Cassandra workshop! In this two-hour workshop, we show how to combine a scalable database, Apache Cassandra™, with a powerful real-time streaming graph processor, Quine.
Duration : 2 hours
Level : Intermediate
DataStax
Developers
Hands-On: Real-time
graph ETL for modern data pipelines with
Quine & Apache Cassandra®

GitPod

01

Introduction

02

Quine streaming graph

03

Use Cases

04

Architecture

05

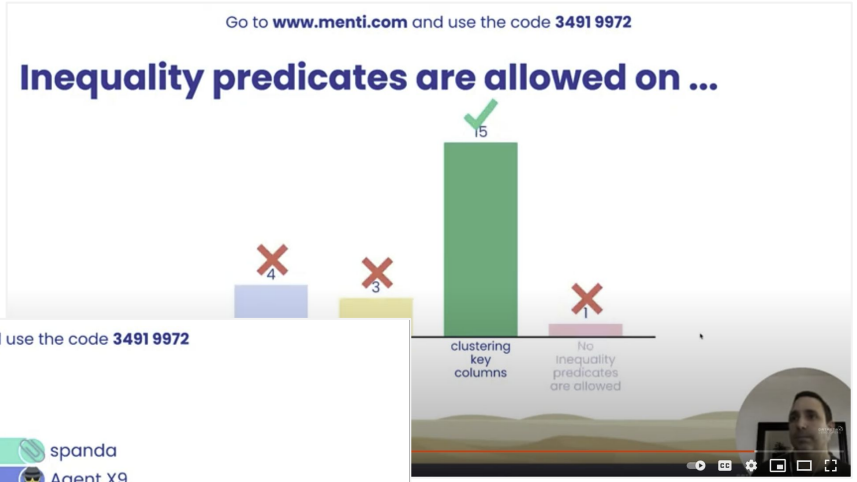
Hands on

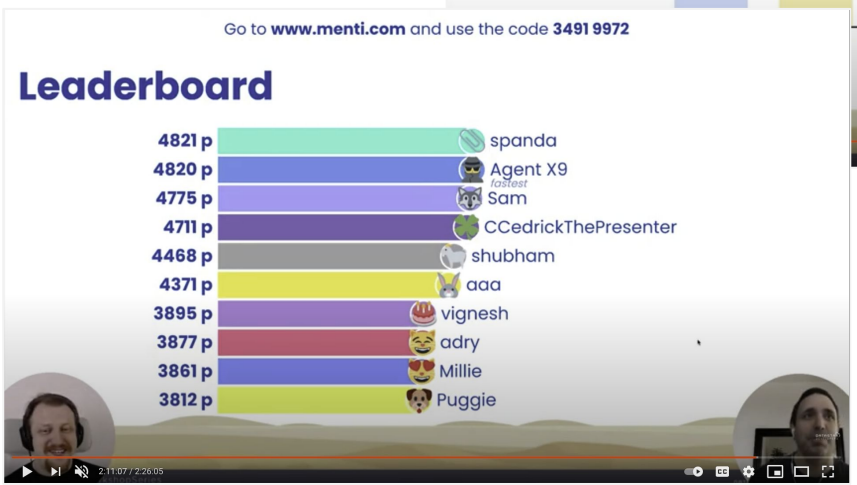
06

What's next?

Quiz, Homework, Next week

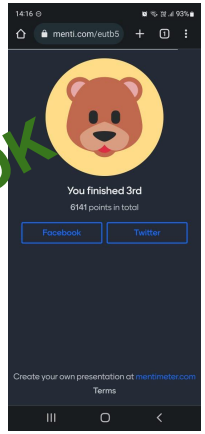
Agenda



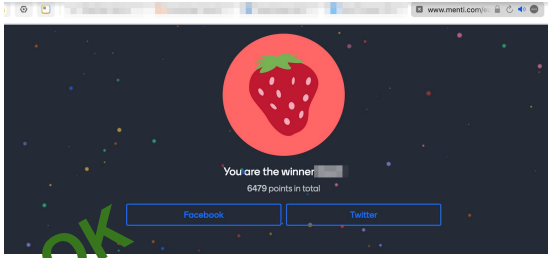


Play with us with Menti.com (new TAB)

SWAG WINNERS 🎉



OK

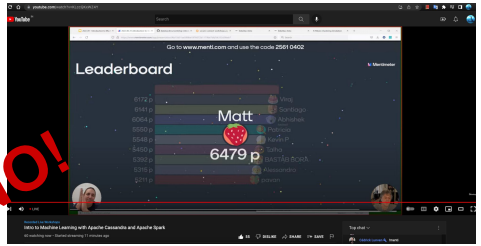


OK

Congratulations to 1st, 2nd and 3rd place on the Menti quiz!

To claim your prize:

Take a screenshot of your Menti screen



NO!

Fill the form at dtsx.io/workshop-swag

Homework

[Complete hands-on]

1. Modify an existing recipe to do something new, and take SCREENSHOT(s).
2. Send the SCREENSHOT(s) and recipe YAML.
3. Email to Aaron and Michael.

aaron.ploetz@datastax.com,
michael@thatdot.com

Subject: Quine Homework

Cassandra Days ... in-person events are back!

- ~~Berlin~~ ~~Sept 20~~
- ~~London~~ ~~Oct 11~~
- ~~Amsterdam~~ ~~Oct 13~~
- Santa Clara Nov 10
- Seattle Nov 10 (**thatDot/Quine speaker**)
- Houston Nov 10 (**thatDot/Quine speaker**)
- Hanoi Nov 8
- Jakarta Nov 10
- Singapore Nov 15

Find out more and register at https://www.datastax.com/events

KubeCon CloudNativeCon

North America 2022

OCTOBER 24 – 28

DETROIT, MICHIGAN

REGISTER VIEW THE SCHEDULE

- **Training day March 12**
- **In-person**
- **Virtual**
- **CFP and Reg coming soon**

Meet with us. Win a Tesla.

Let's talk DBaaS built on Apache Cassandra®.

BOOK A MEETING NOW

*3-year Tesla Model 3 lease. NO PURCH NEC. Ends 12/31/22.

DataStax

intel. XEON

bit.ly/datastax_tesla

*Terms & conditions apply

Thank You!

Quine
+
Cassandra
WORKSHOP
DataStax Developers