



Building lightning-fast data apps with Firebolt and Cube

Luka Lovosevic
Solution Architect @ Firebolt

Igor Lukanin
Head of Developer Relations @ Cube

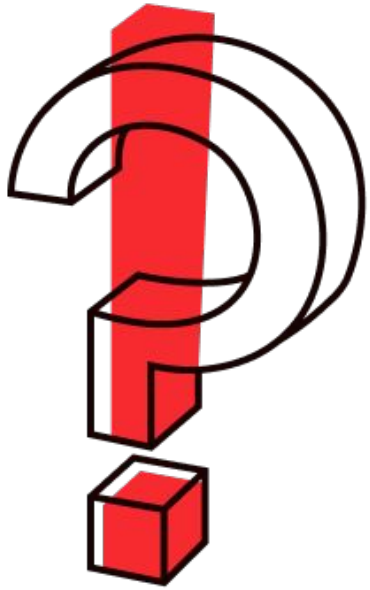
F The rise of 'Data Apps' and instant analytics:

A few seconds is too slow



According to **research by Google**, 53% of mobile website visits are abandoned if a site takes longer than 3 seconds to load.

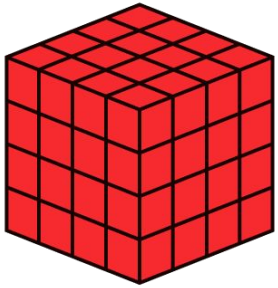
Data Apps?



- Packaged consumer-grade analytics experiences
- Fast, robust and always up, like any other app
- An evolution of traditional analytics
- Delivered by software engineering teams

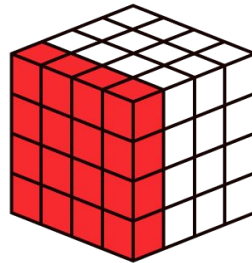


Types of Data Apps



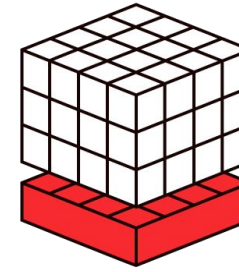
Products

The data and the experience around it is the product



Modules

Data experiences that are value-adds to existing products

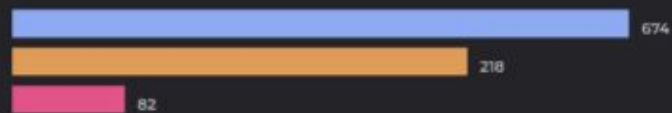


Internal tooling

Purposeful analytics experiences intended to improve operations and efficiencies

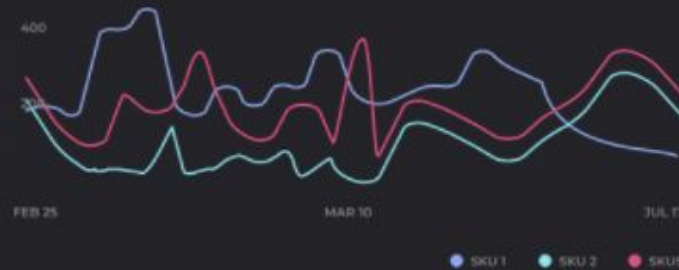
Data app examples: **Product**

ORDER ACCURACY



● LOS ANGELES ● HOUSTON ● NEW JERSEY

MISSING ITEMS



WARRANTY CLAIMS TRACKER



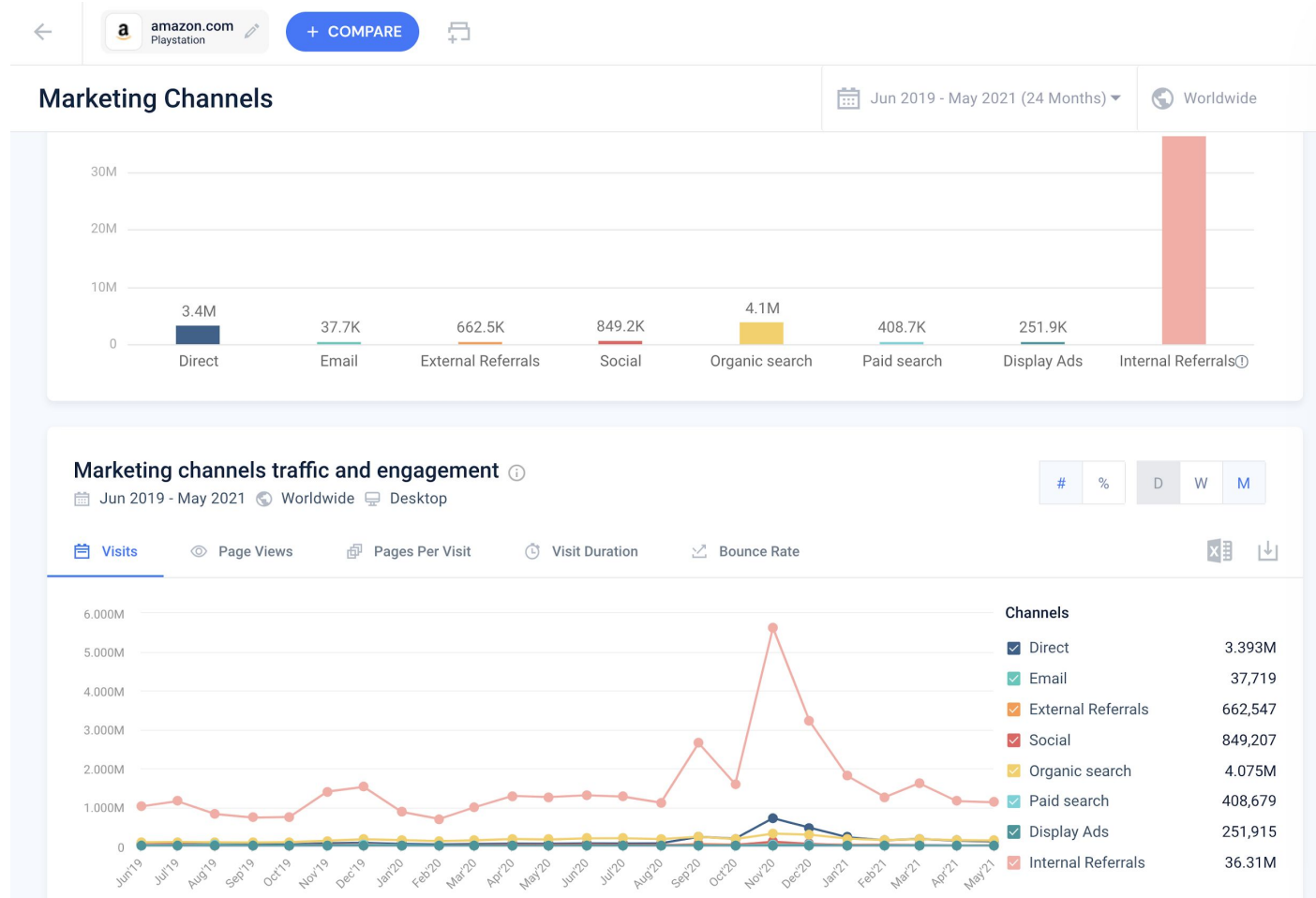
Product SKUs

182	182
TS44GR	TS44WH
674	10
TS44BK	TS42BK

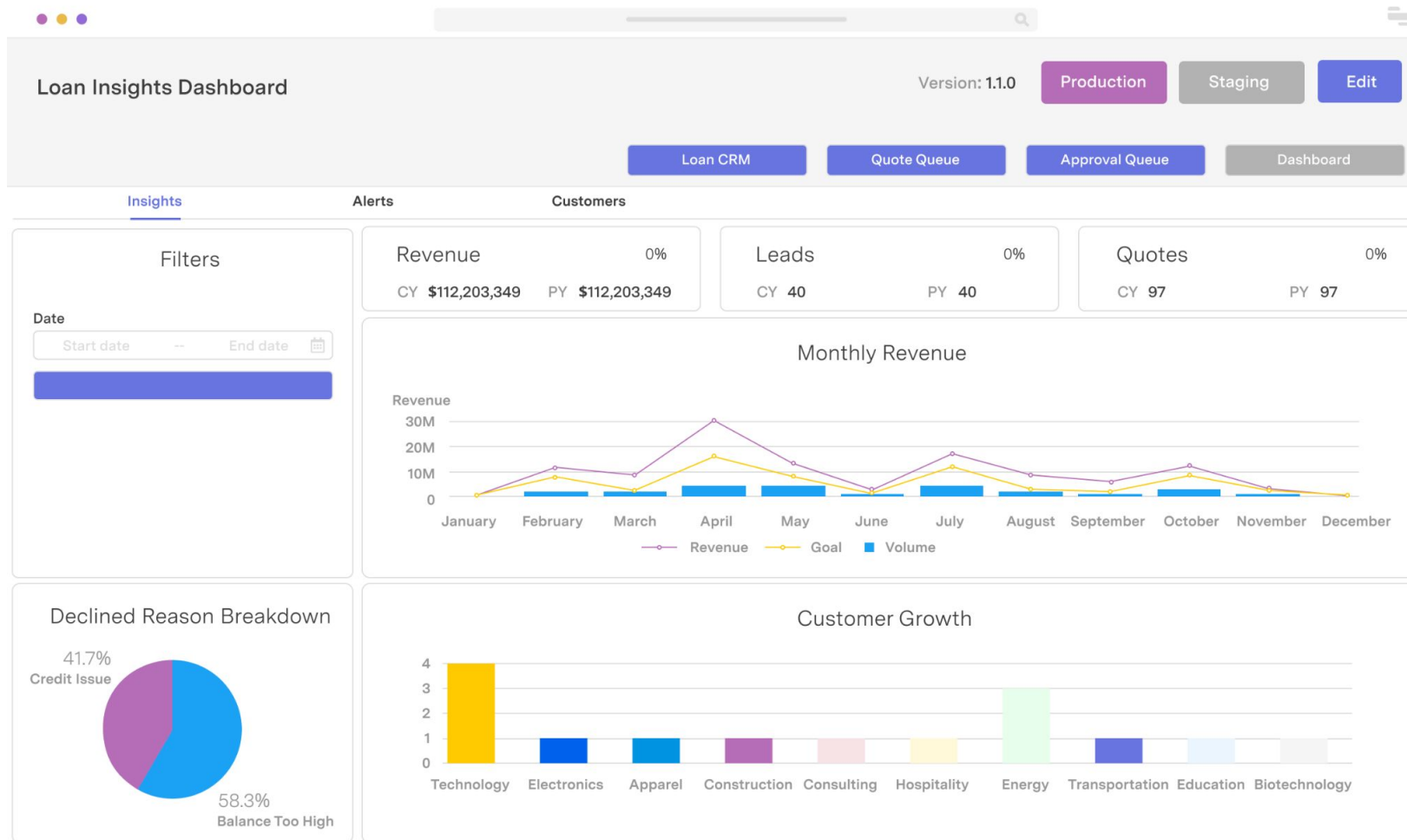
DAMAGED GOODS



F Data app examples: Module



Data app examples: Internal Tooling



A great decade for Cloud Analytics... met by **new challenges**

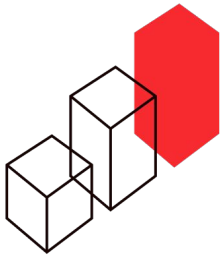


More data, more problems

The variety and volume of data that can be collected is constantly on the rise.

- **TB is the new GB**

If that's not the case for you, just wait 😊



The rise of 'Data Apps' and instant analytics

Analytics is evolving from traditional BI to instant, operational and customer-facing data experiences.

- **A few seconds is too slow**

Users expect sub-second, consumer grade experiences

A great decade for Cloud Analytics... met by **new challenges**

Painful tradeoffs limit the analytics experience

There is no choice but to compromise



Aggregate data into summary tables in lengthy jobs, losing granularity and freshness.



Analyze **smaller time windows**, losing depth.

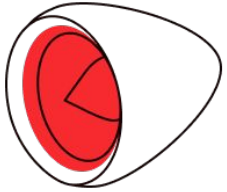


Accept **slow dashboards** and frustrated users.



Glue together multiple point solutions in **complex engineering projects**.

What makes a data app that end-users **love**?



It's all about performance

Your data warehouse performance = your product performance.
Queries must integrate within the low-latency snappy experience



Flexible use of a well-defined data model

Data apps often enable users to slice and dice data in different ways, which requires flexibility from the warehouse side. On the other hand, schema is not changing very often.



Big data

Or at least enough data that performance becomes a concern - now or in the future

Firebolt: The Cloud Data Warehouse for builders of **next-gen** analytics experiences



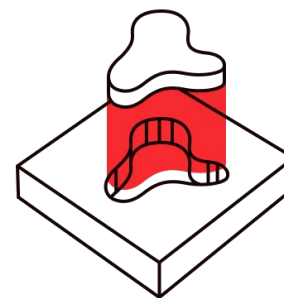
Built for developers & data engineers

Built to fit into the modern data & engineering stack, so that you can **deliver data apps** quickly & reliably.



Performance & Scale

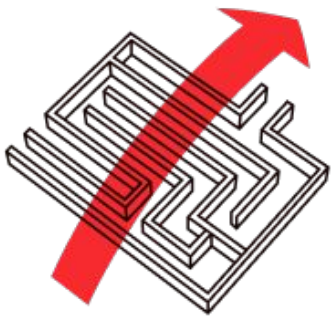
Deliver **sub-second analytics experiences at TB+ scale**, while keeping HW resources low and without painful tradeoffs.



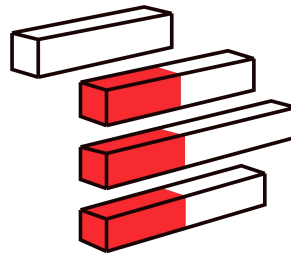
A modern cloud-native architecture

Easily isolate & match the right resources to the right workload over a **decoupled storage/compute** architecture.

F Built for data app **speed**



The goal: Move & scan
much less data



The accelerator:
Sparse Indexing



The building block:
TripleF Files

F Built for data app **speed**



The building block:
TripleF Files

```
CREATE TABLE fact_round (  
    event_id,  
    sdfsd,  
    ...  
) PRIMARY INDEX event_id, date, customer_id
```

Even faster with more index types

Aggregating Index

- Great for repeating workloads such as dashboards/reports
- Defined on the table once, and always stays in sync. Add as many as needed.
- Indexed queries can use smaller cluster sizes due to lower hunger for CPU

```
CREATE AGGREGATING INDEX idx_agg_rounds ON fact_round
(  
    game_code,  
    player_code,  
    currency_code,  
    count(distinct round_id),  
    sum(credit),  
    avg(credit),  
    sum(debit),  
    sum(total_events)  
)
```

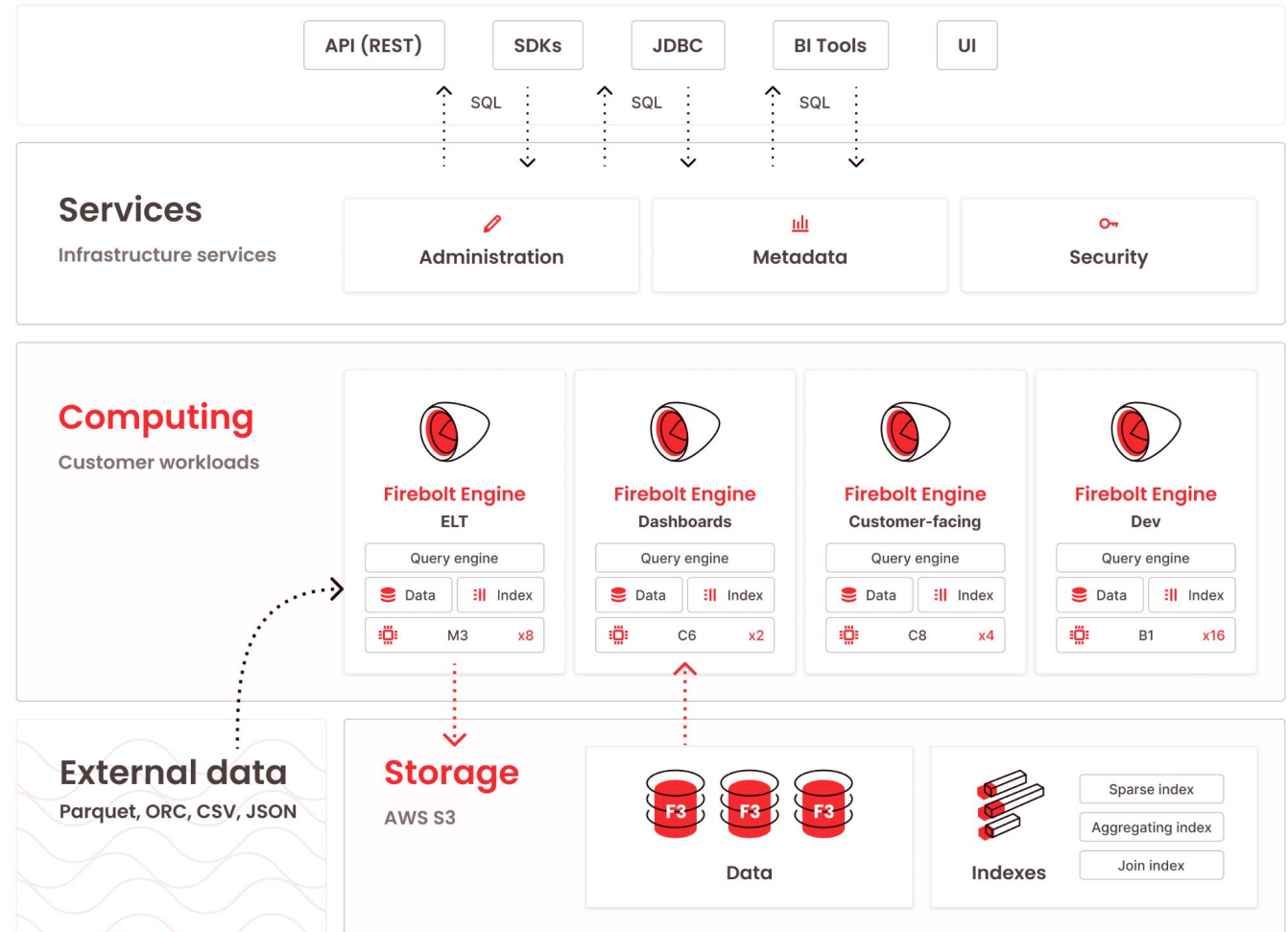
Join Index

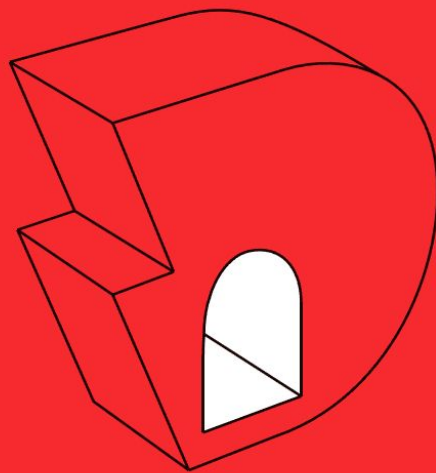
- No need to denormalize schemas to avoid join slowdowns
- Join indexes are RAM stored objects that accelerate joins dramatically
- Can work together with agg. index

```
CREATE JOIN INDEX idx_join_games ON dim_games
(  
    game_code,      -- join column  
    game_studio,    -- dim column  
    game_currency   -- dim column  
)
```

A modern architecture for tomorrow's data challenges

- Engines are tunable so that you can match your workload with the right compute for optimal price-performance
- Robust API, python and NodeJS SDK. Native integration to top BI tools
- Indexing which allows for sub-second analytics
- ANSI Compliant SQL language





Firebolt Demo

Querying Wikipedia at speed

 Building lightning-fast data apps with Firebolt and Cube

Transition into Cube slides

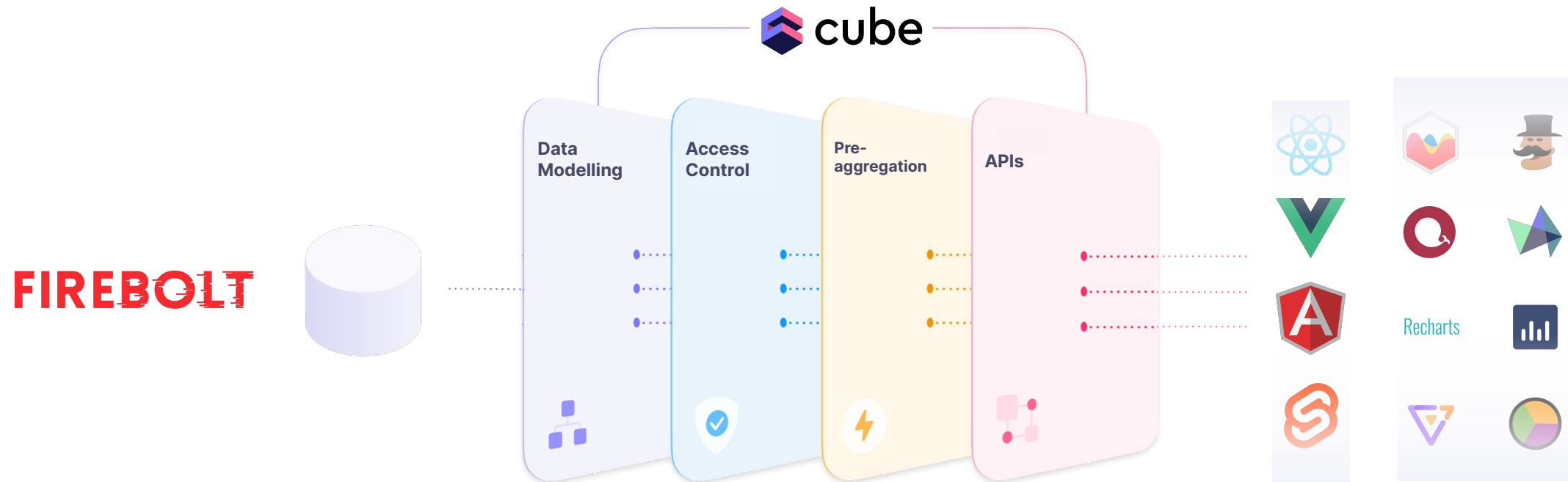
Cube: Headless BI platform

Cube is a powerful tool for...

- **consuming** data from **modern data stores**,
- **organising** it into consistent definitions, and
- **delivering** it to every application.

Cube's mission is making data consistent and accessible.

Layers of Headless BI



Consistency

Cube provides a centralized place for...

- **metrics definitions** (so they are reused by all data apps)
- **access control rules** (so they apply to all data apps)
- **query cache** (so it's consistent across all data apps)

Consistency — Data modeling

Metrics are defined in a declarative, LookML-like data model.

cube cloud

Deployments / bm_firebolt_tutorial VPCs Docs IG

master Enter Development Mode

API is ready (Build #1980) Copy URL

Schema

Files Changes

schema

BmCubeOutliers.js

cube.js

package.json

BmCubeOutliers.js

```
1 cube('Outliers', {
2   sql: `
3     SELECT
4       *,
5       UPPER(SPLIT_PART(wiki_code, '.', 1)) AS region,
6       CONCAT('https://', wiki_code, '.org/wiki/', article_title) AS url
7     FROM (
8       SELECT * FROM dev.bm_cube_outliers
9       UNION
10      SELECT * FROM dev.second_load_bm_cube_outliers WHERE log_date >= '2022-06-07'
11    )
12   `,
13
14   measures: {
15     count: {
16       type: 'count'
17     },
18
19     dailyTotal: {
20       sql: 'daily_total',
21       type: 'sum'
22     }
23   },
24
25   dimensions: {
26     wikiCode: {
27       sql: 'wiki_code',
28       type: 'string'
29     },
30   },
```

Consistency — Data modeling

Metrics are defined in a declarative, LookML-like data model.

Measures, dimensions, joins, etc. are reused by all data apps.
Data model respects multitenancy and can be dynamic.

- ✓ No duplicated effort.
- ✓ No inconsistency.

Consistency — Access control

Access control rules are defined in code.

cube cloud

Deployments / bm_firebolt_tutorial VPCs Docs IG

master Enter Development Mode

API is ready (Build #1980) Copy URL

Schema

Files Changes

/

schema

BmCubeOutliers.js

cube.js

package.json

BmCubeOutliers.js X

cube.js X

```
1 module.exports = {
2   queryRewrite: (query, { securityContext }) => {
3     // if (!securityContext.wikipedias) {
4     //   throw new Error('Wikipedias in the Security Context are mandatory!!');
5     // }
6
7     if (securityContext.wikipedias) {
8       query.filters.push({
9         member: 'Outliers.region',
10        operator: 'in',
11        values: securityContext.wikipedias,
12      });
13    }
14
15    return query;
16  }
17};
```

Consistency — Access control

Access control rules are defined in code.

Rules can inspect a query and meta data, add mandatory filters, or completely rewrite the query. Meta data is provided with each query as JSON Web Token payload.

 Universal upstream access control

Accessibility

Cube provides a set of APIs:

- **REST API** with JavaScript/React/Vue/Angular libraries
- **GraphQL API**
- **SQL API**

Regardless of the API flavor, queries yield consistent results.

 Any downstream tool can consume the data.

Accessibility — REST API

Semantic JSON-based queries secured with JSON Web Tokens.

Overview

aws US East (Ohio)

REST API

<https://mutual-crayfish.aws-us-east-2.cubecloud>

SQL API

With this new API, Cube can now function as a headless compatible querying and visualization tool.

GraphQL API beta

<https://mutual-crayfish.aws-us-east-2.cubecloud>

Resources & Logs

Cluster

- ✓ Cube API
- ✓ Redis
- ✓ Cube Refresh Workers
- ✓ Cube Store

Connecting to REST API

Learn more about [REST API](#) and [integration with frontend frameworks](#).

Terminal	Vanilla JS	React	Angular	Vue
----------	------------	-------	---------	-----

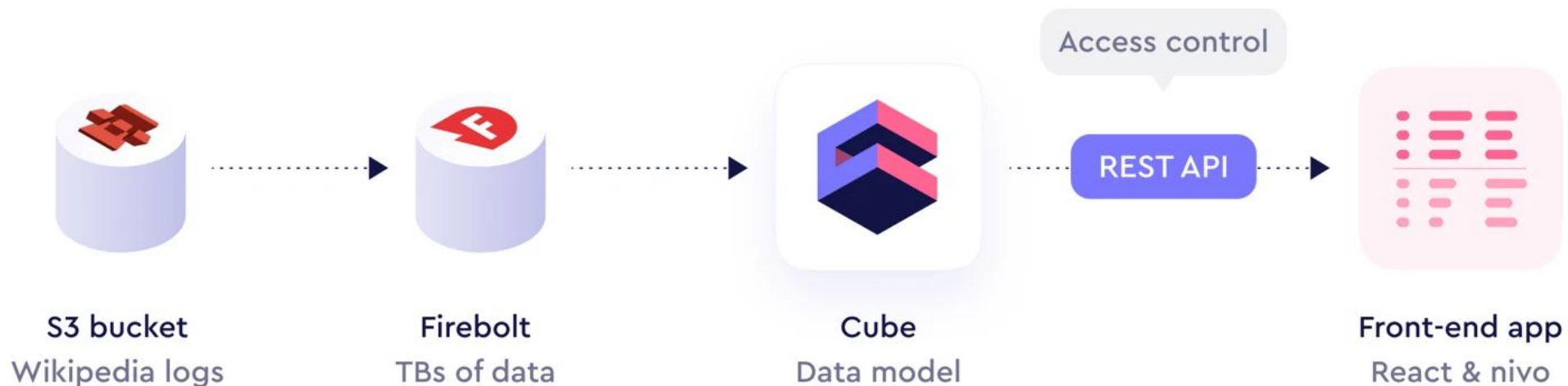
Init Cube API:

```
import cubejs from '@cubejs-client/core';
const cubejsApi = cubejs({
  'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpYXQiOiJlE2NjM5NDIyN'.
  { apiUrl: 'https://mutual-crayfish.aws-us-east-2.cubeclouda'
});
```

Get a result set:

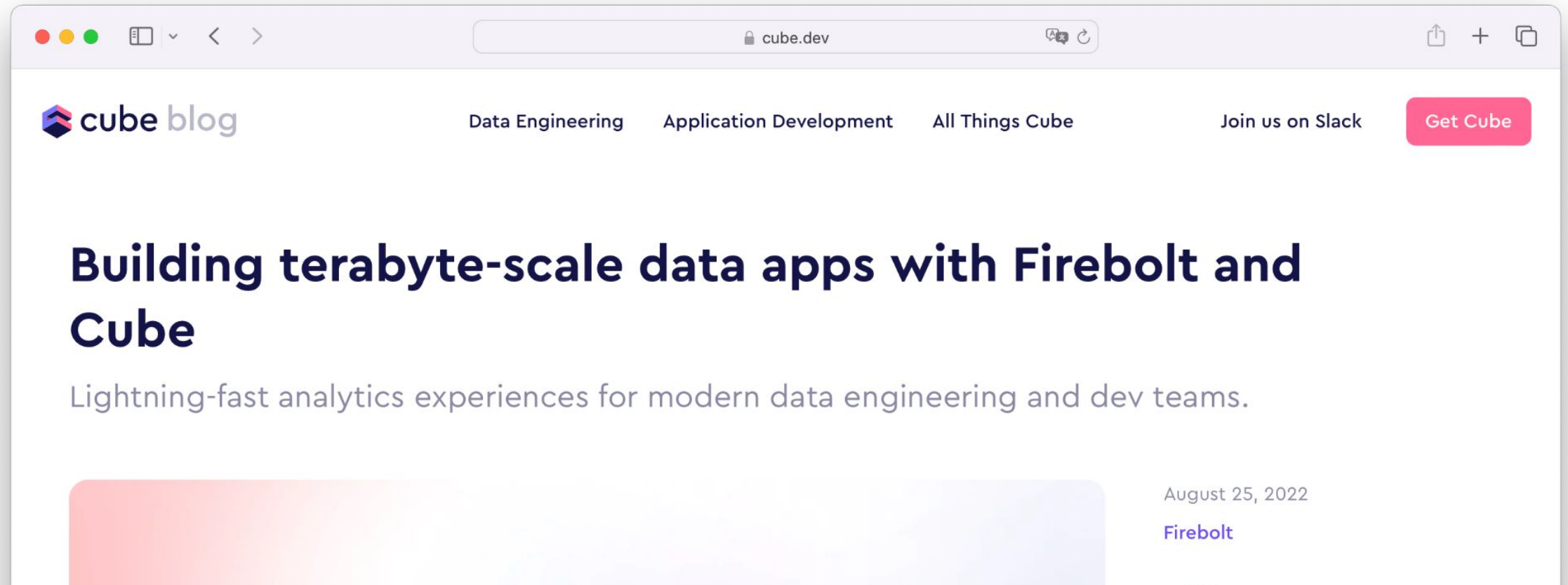
```
const resultSet = await cubejsApi.load({
  "measures": ["Outliers.count"]
});
```

- Use **Firebolt** to analyse logs from Wikipedia—~**4TB**/yr
- Identify sudden spikes in traffic corresponding to real-world events
- Use **Cube** for queries and access control
- Build a **React + nivo** front-end
- Be ***fast***



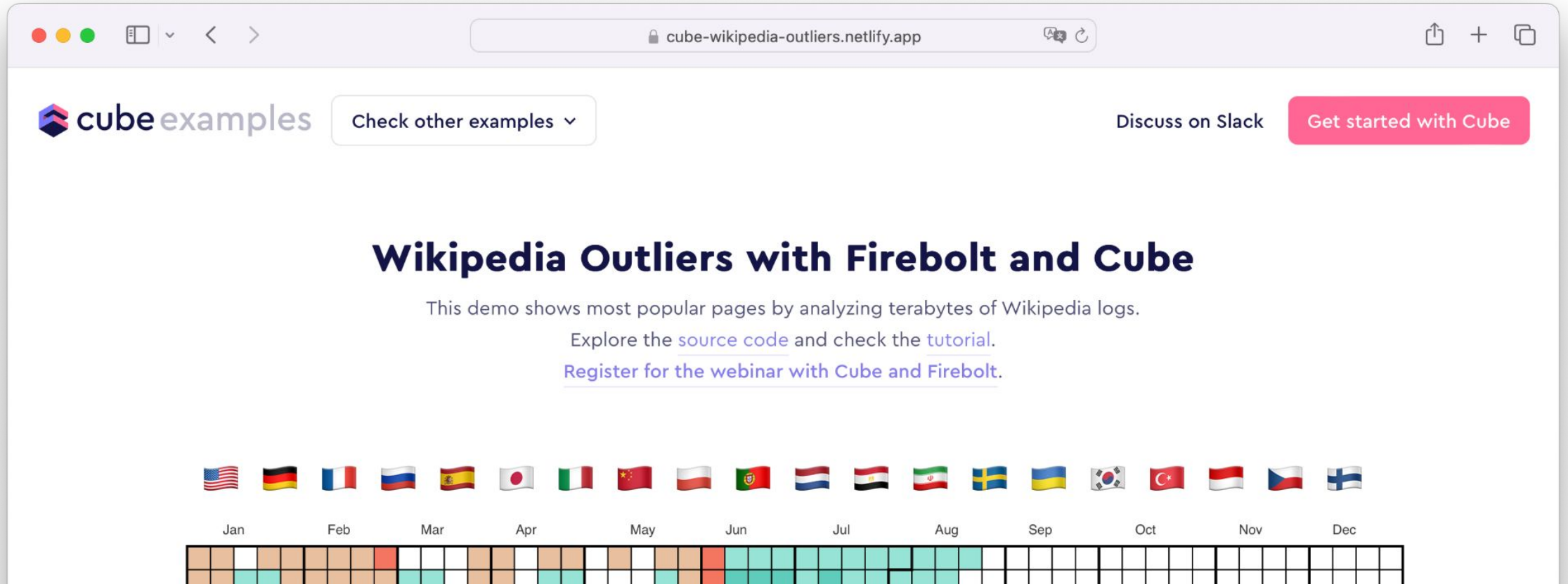
Building terabyte-scale data apps

cube.dev/blog/building-data-apps-with-firebolt-and-cube



Wikipedia Outliers data app

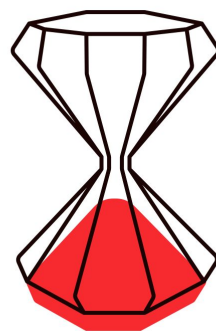
cube-wikipedia-outliers.netlify.app



Cube demo time!

We will observe:

- **Cube deployment** in Cube Cloud
- **Front-end application** built with React and nivo
- **Source code**



Thank you for your time

FIREBOLT +  **cube**