

Data Movement for Data Developers

Deliver data generated by your production applications to analytics users at scale, without sacrificing quality.

Upsolver

Who we are

The only self-serve cloud data movement service for high-scale workloads (*transaction, streaming and AI*)

Upsolver's decoupled architecture makes it an ideal solution for ingesting and processing high volume and complex data, efficiently and reliably.

Certifications



Partners



CONFLUENT



dbt

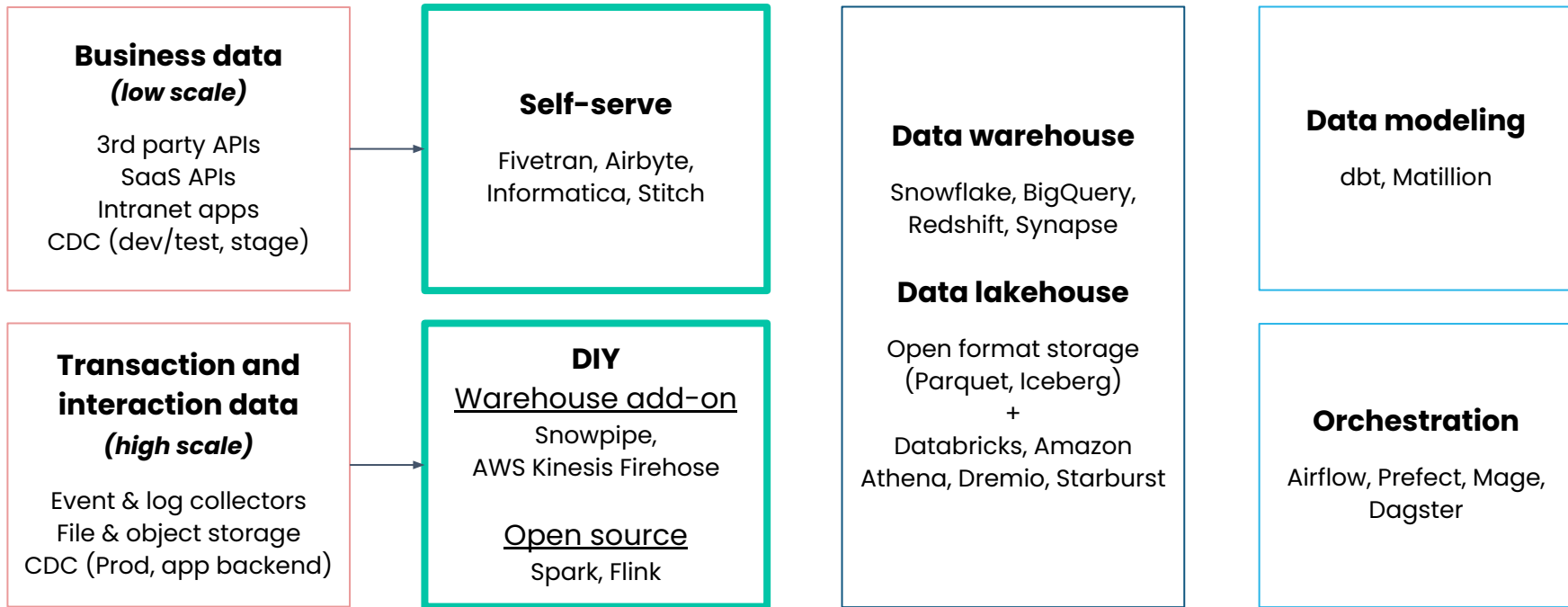
Proven at scale

8% of customers	>1PB/mon
26% of customers	>100TB/mon
54% of customers	>10TB/mon
82% of customers	>1TB/mon

Ingestion in the modern data stack

The modern data stack (MDS) makes it simple for companies to become data-driven

Ingestion



Ingestion in the modern data stack

But for high scale, existing ingestion solutions fall short

Ingestion

Self-serve

Fivetran, Airbyte,
Informatica, Stitch

Challenge

Fails or becomes prohibitively
expensive at scale

DIY

Warehouse add-on

Snowpipe,
AWS Kinesis Firehose

Open source

Spark, Flink

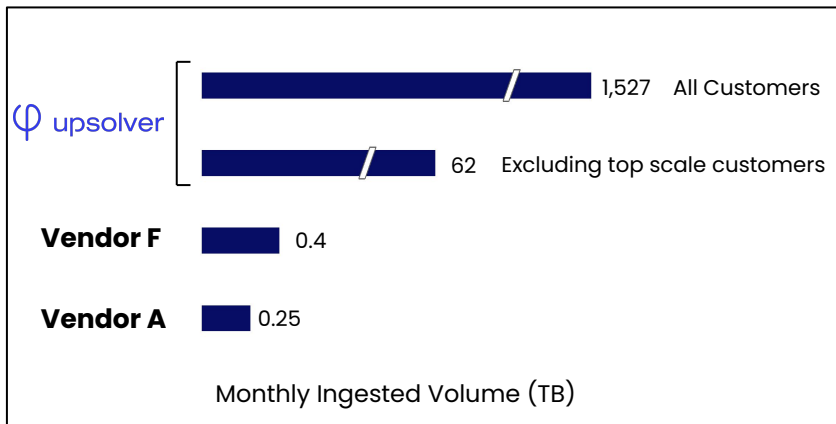
Challenge

Difficult to build, manage and optimize,
leading to tech debt & downtime

Self-serve ingestion tools are **NOT** built/priced for scale

Because they focus on long tail connectors (SaaS, APIs) for small business data

Popular self-serve tools only handle a **tiny** fraction of your data volume



Self-serve tools cost you on average **10X more** than Upsolver or DIY

Scale	Vendor F	Vendor A	Upsolver
1 TB	\$22,000	\$5,000	\$4,225
10 TBs	\$50,000	\$20,000	\$6,250
100 TBs	\$500,000	\$200,000	\$26,500

Cost per month

Assumptions:

- Append-only ingestion, 1KB per row
- Upsolver list price - \$4,000 + #TBs*\$225

DIY is a bottleneck to being data-driven

Takes away focus and resources from high value business initiatives

Data initiatives orgs want to invest in

AI / ML

Personalization

Data apps

Real-time insights

Customer-360 & CDP

Where time is spent with DIY

Integrating lots of tools

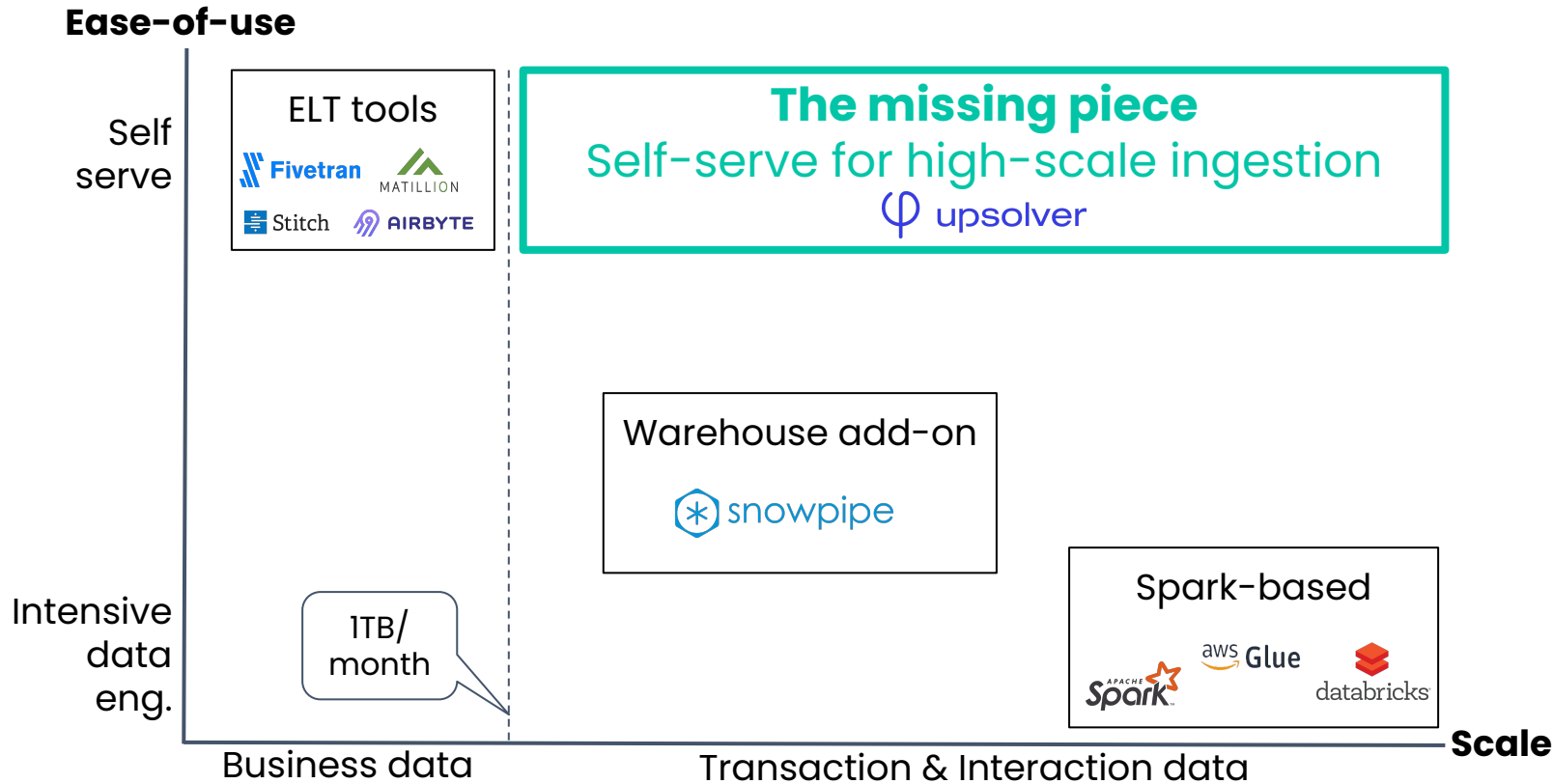
Writing & testing scripts

Building & testing orchestration DAGs

Fixing quality problems

Manually evolving schemas

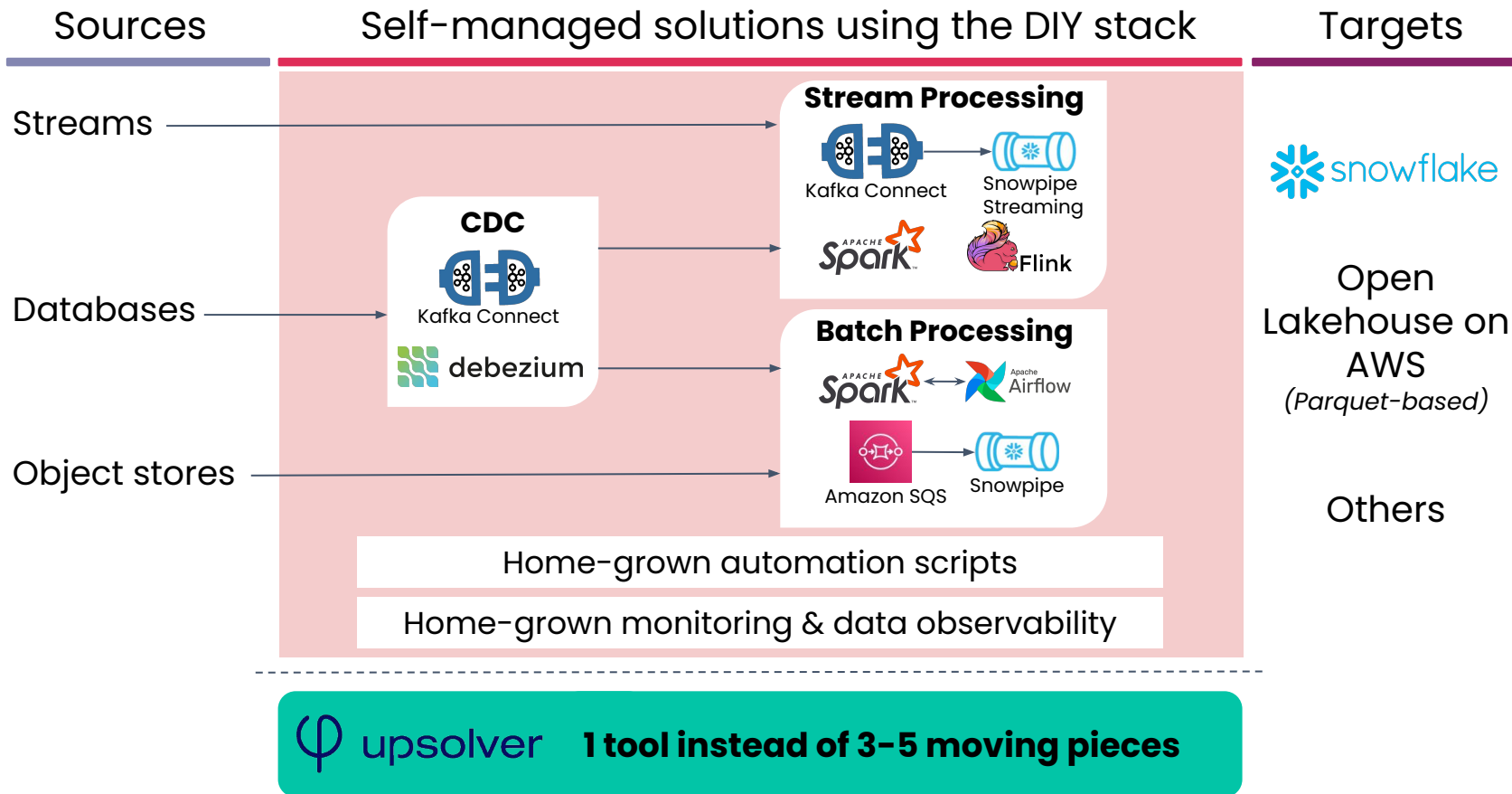
Upsolver fills the tooling gap for high-scale ingestion



Upsolver capabilities that remove DIY complexities

	DIY – Open source (Spark-based)	DIY – DW add-on (Snowpipe)	Self-serve (Upsolver)
E(T)L in a single tool	No	No	Yes
No-code and low-code dev experiences	No	No	Yes
Protects analytics and ML from data incidents	No	No	Yes
Improves data quality – discover, resolve, prevent	No	No	Yes
Outcome	Months to implement, Constant break-fix and ops headaches		Hours to implement, near-zero ops

1 E(T)L in a single tool



2 No-code & low-code dev experiences

1

Set up your source

- Use an existing connection
 - upsolver_kafka_samples
- Create a new connection

Select a topic to ingest

Select the source events content type

Automatic

Sample Events

user_info

Sample Size

100

Paused Size

100

```
{ "id": "0", "data": { "Event_time": 1683531485840 } }
{ "id": "1", "data": { "Event_time": 1683531485840 } }
{ "id": "2", "data": { "Event_time": 1683531485840 } }
```

2

Set up your target

- Use an existing connection
 - snowflake_target_conn
- Create a new connection

Select a schema

DEMO

New table name

USER_INFO

3

Configure the ingestion

Name your job

user_info_to_USER_INFO

How often do you want to update the target? (writing interval)

1 Minute Hour

Which events to ingest?

Start from now

Prevent duplicate events (deduplication)

Select fields for deduplication key

user_id X

Deduplication window

1 Minute Hour

Schema configuration

As soon as the schema changes, the newly ad

Search

- x T address
- ✓ T credit_card
- ✓ T first_name
- ✓ T last_name

4

Review and run job

```
CREATE SYNC JOB upsolver_kafka_samples_to_upsolver_snowflake
CREATE_TABLE_IF_MISSING = true
START_FROM = BEGINNING
CONTENT_TYPE = AUTO
DEDUPLICATE_WITH = (COLUMNS = (orderid) WINDOW = 1 HOURS)
COLUMN_TRANSFORMATIONS = (customer.email = MD5(CAST(customer.email AS STRING)))
WRITE_INTERVAL = 1 HOURS
EVENT_TIME_COLUMN = UPSOLVER_EVENT_TIME
ADD_MISSING_COLUMNS = true
AS COPY FROM KAFKA upsolver_kafka_samples TOPIC = 'orders'
INTO SNOWFLAKE upsolver_snowflake.SUMMIT.ORDERS;
```

Copy SQL

Previous

Edit in Worksheet

Run

Code is simple and therefore easy to understand, test and manage

Dev options

Upsolver GUI

Upsolver Worksheet

dbt

CLI

SDK

load_user_events

Running

Up to date

0

14 days ago

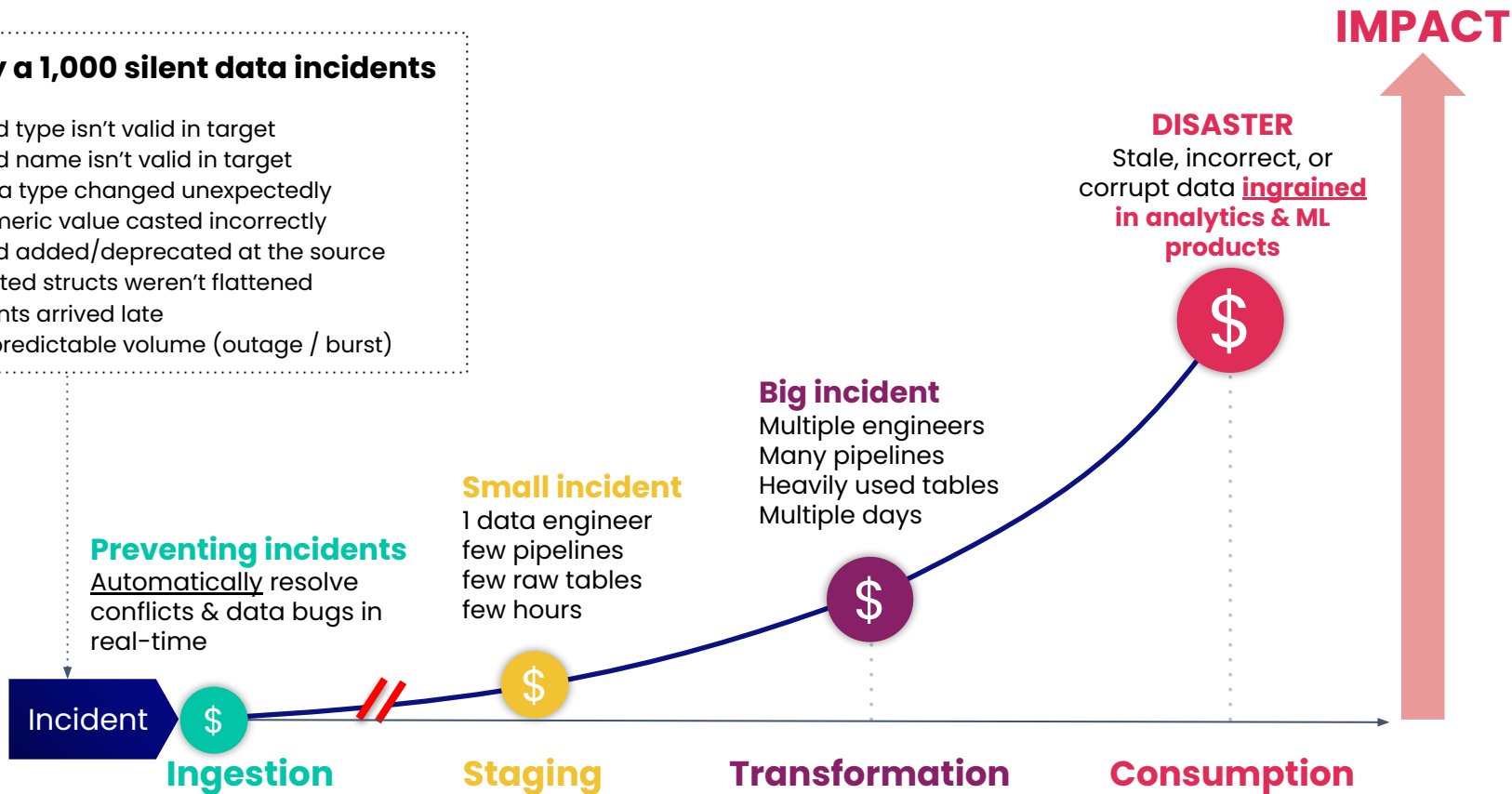
3 Protects analytics and ML from data incidents

Death by a 1,000 silent data incidents

- Field type isn't valid in target
- Field name isn't valid in target
- Data type changed unexpectedly
- Numeric value casted incorrectly
- Field added/deprecated at the source
- Nested structs weren't flattened
- Events arrived late
- Unpredictable volume (outage / burst)

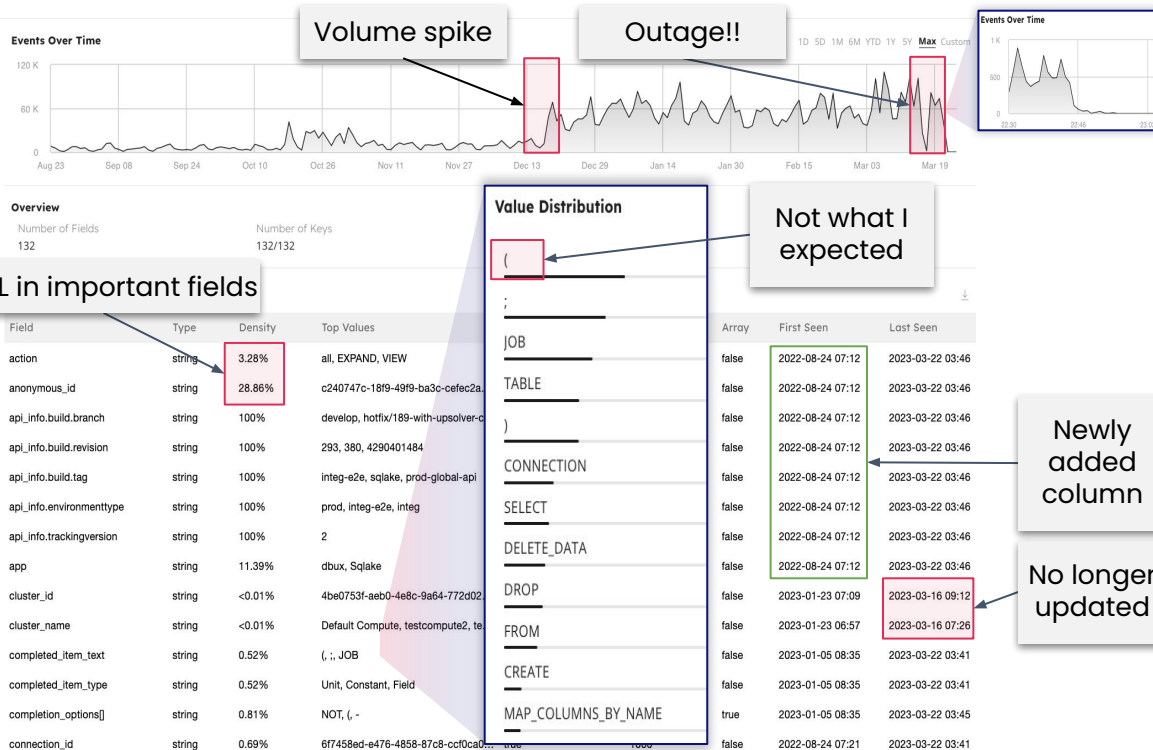
Preventing incidents

Automatically resolve conflicts & data bugs in real-time



Improves data quality – discover, resolve, prevent

Discover using built-in data observability



Resolve & Prevent

```
CREATE SYNC JOB ingest_from_kafka_to_snowflake
  CONTENT_TYPE = JSON
  START_FROM = BEGINNING
  CREATE_TABLE_IF_MISSING = TRUE
  WRITE_INTERVAL = 1 MINUTE
  DEDUPLICATE_WITH = (COLUMNS = (orderid) WINDOW = 1 MINUTES)
  COLUMN_TRANSFORMATIONS = (customer.email = MD5(customer.email))
  AS COPY FROM KAFKA upsolver_kafka TOPIC = 'orders'
  INTO SNOWFLAKE upsolver_snowflake.production.ORDERS_DATA
  WITH EXPECTATION nonzero_nettotal
  CHECK netttotal <> 0 ON VIOLATION WARN
  WITH EXPECTATION orderid_notnull
  CHECK orderid IS NOT NULL ON VIOLATION DROP
  WITH EXPECTATION invalid_states
  CHECK LENGTH(customer.address.state) = 2 ON VIOLATION WARN;
```

Resolve
incorrect data
using replay

Newly
added
column

No longer
updated

Prevent incorrect
data from entering
the warehouse using
quality expectations

Success over Self-Service ELT



Cyber security unicorn

CDC from PostgreSQL to Snowflake

60 TBs / month

Replaced Fivetran and Spark



Talent acquisition CRM

CDC from PostgreSQL to Snowflake

6 TBs / month

Replaced Fivetran



"Upsolver is solving a problem (CDC) at scale that I don't want to think about. The fact that it just works and people don't complain is a win."

Yuji Xie, Analytics Lead

Success over DIY add-on tools

CENTENE®
Corporation

Healthcare provider, Fortune #26

Kafka to Snowflake

4 TBs / month

Self-service: 100+ sources, 60+ users

Replaced AWS Glue and Snowpipe

avantis

App monetization platform

Kafka to Snowflake

160 TBs / month

Eliminated data eng. bottleneck

Replaced AWS Glue and Snowpipe

CENTENE®
Corporation



"Upsolver is like the 'easy button' for Snowflake.

We ingest data from our Kafka streams, process it for different use cases, and deliver it, all while observing how our schema and data are changing in real time."

Alexander Adam, Manager, Data Lake Cloud ETL

Success over DIY open source



App monetization platform

Kafka to AWS ecosystem

50 PBs / month

Serving 100+ data analysts/scientists

Replaced Spark on Amazon EMR



Manufacturing, global 2000

IoT to AWS

52 TBs / month

Self-serve for multiple IoT initiatives

Replaced AWS Glue



"We want to minimize the time our engineering teams, including DevOps, spend on infrastructure and maximize the time spent developing features. Upsolver has saved thousands of engineering hours and significantly reduced total cost of ownership"

Seva Feldman, Vice President of Research and Development

The only self-serve cloud data movement service for high-scale workloads

What you get with Upsolver

- SaaS or AWS VPC
- E(T)L in a single tool
- Supports popular CDC, streaming and object store sources
- Outputs to Snowflake, Redshift and your Lake on Amazon S3
- Simple to use no-code and low-code dev experience
- Integrates directly with your development workflows (Github, CI/CD)
- Automatic schema evolution and type resolution
- Built-in, real-time data and pipeline observability

Priced to scale starting* at \$150/TB of data ingested

*<https://www.upsolver.com/pricing>

