



Wolfram „Wolle“ Wingerath

Data Validation at Scale: Managing Data Quality in Complex Data Pipelines

Data Analytics & Science



code.talks

Data **Validation** at Scale

Managing Data Quality in Complex
Data Pipelines

code.talks 2023, Hamburg, Germany

September 14, 2023

Wolle

Material Available at <https://wolle.science>

I Am **Wolle**



Research:

- Stream Processing
- Real-Time Databases
- NoSQL & Cloud Systems
- ...



Practice:

- Web Caching
- Big Data Analytics
- Anger Management
- ...



Talk **Outline**

1

The Importance of Data Validation

Where is data validation integrated into data science pipelines and what is its impact?

2

Data Quality & Constraints

What dimensions of data quality are there and how can they be ensured?

3

Scalability-Related Challenges

Why is data validation difficult in data-intensive domains?

Running **Example**: Web Performance Analysis Pipeline

Collection

Ingestion

Analytics

Reporting



Tracking (RUM)



Felix Gessert. Batching Was Yesterday: Real-Time Tracking & Analysis For 100+ Million Visitors, code.talks (2023)

code.talks 2023, Sept. 14
Thursday, 17:00
(Data Analytics & Science)

- Performance Dashboard
- QA Dashboard
- Real-Time Alerting
- Ad-hoc SQL Interface
- Custom Reporting

W. Wingerath, B. Wollmer, M. Bestehorn, S. Succo, F. Bücklers, J. Domnik, F. Panse, E. Witt, A. Sener, F. Gessert, N. Ritter. Beaconnect: Continuous Web Performance A/B-Testing at Scale, VLDB (2022)

Running **Example**: Web Performance Analysis Pipeline

Collection



Tracking
(RUM)



Ingestion



Analytics



SQL Interface



Reporting



Performance
Dashboard



QA
Dashboard



Real-Time
Alerting



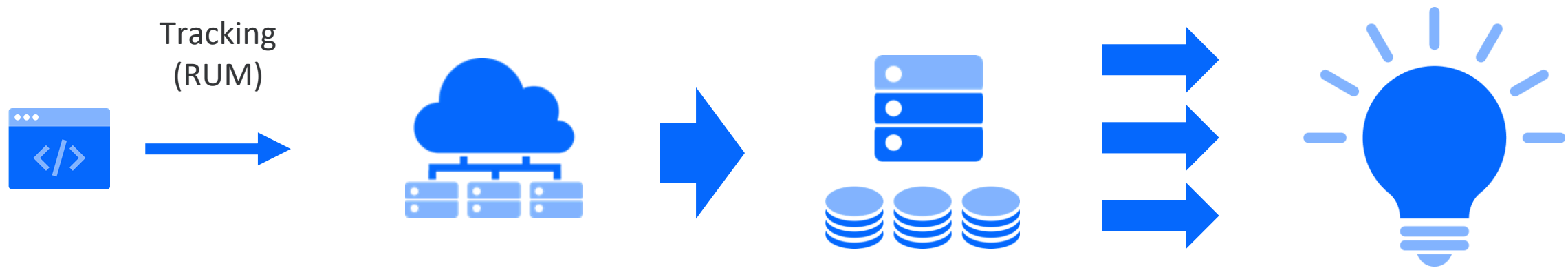
Ad-hoc
SQL Interface



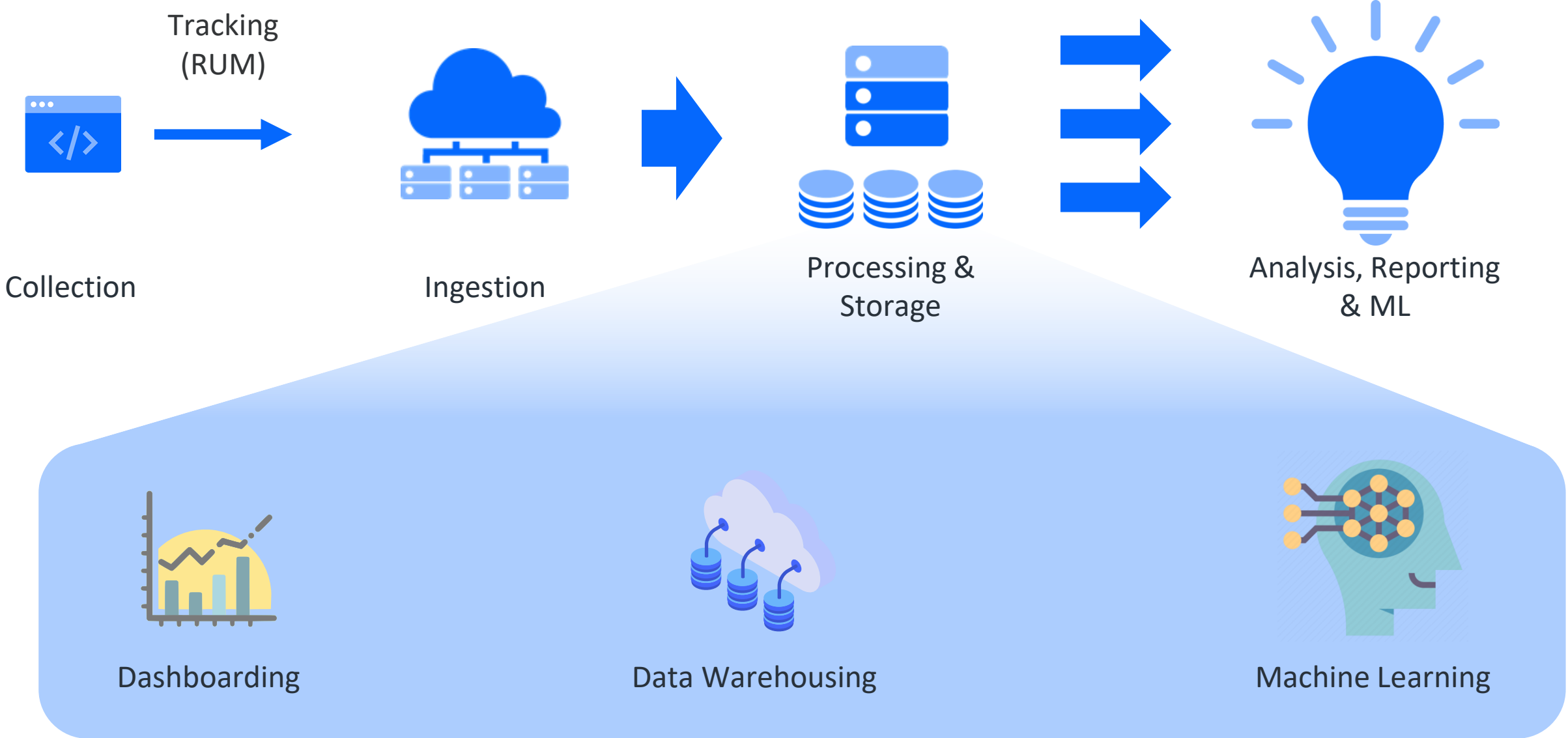
Custom
Reporting

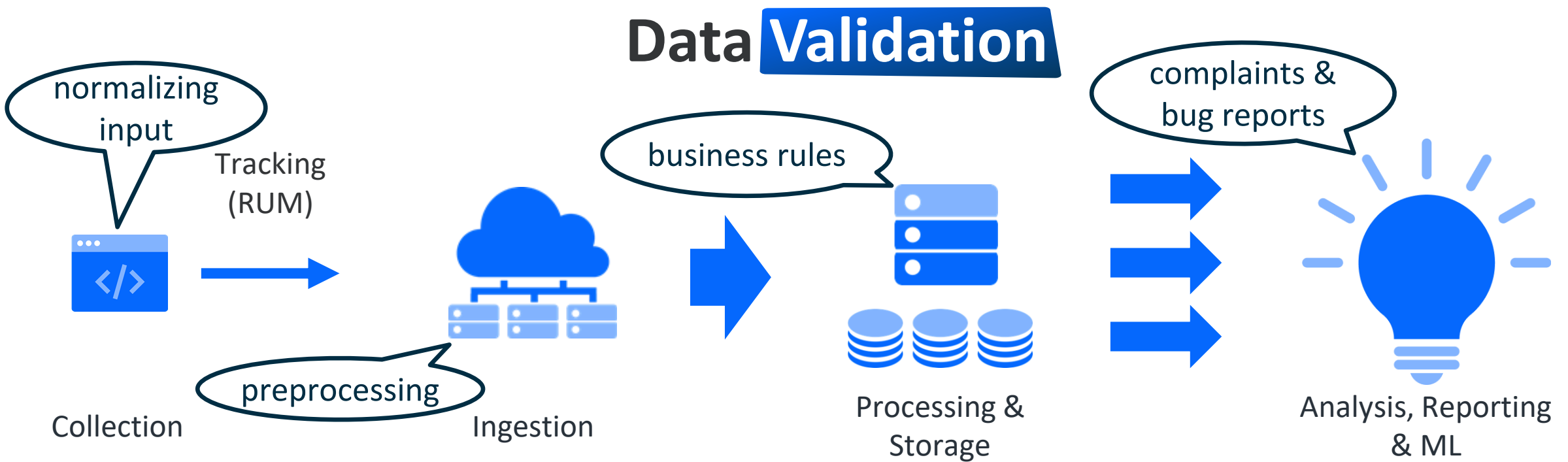
W. Wingerath, B. Wollmer, M. Bestehorn, S. Succo, F. Bücklers, J. Domnik, F. Panse, E. Witt, A. Sener, F. Gessert, N. Ritter. *Beaconnect: Continuous Web Performance A/B-Testing at Scale*, VLDB (2022)

Running **Example**: Web Performance Analysis Pipeline



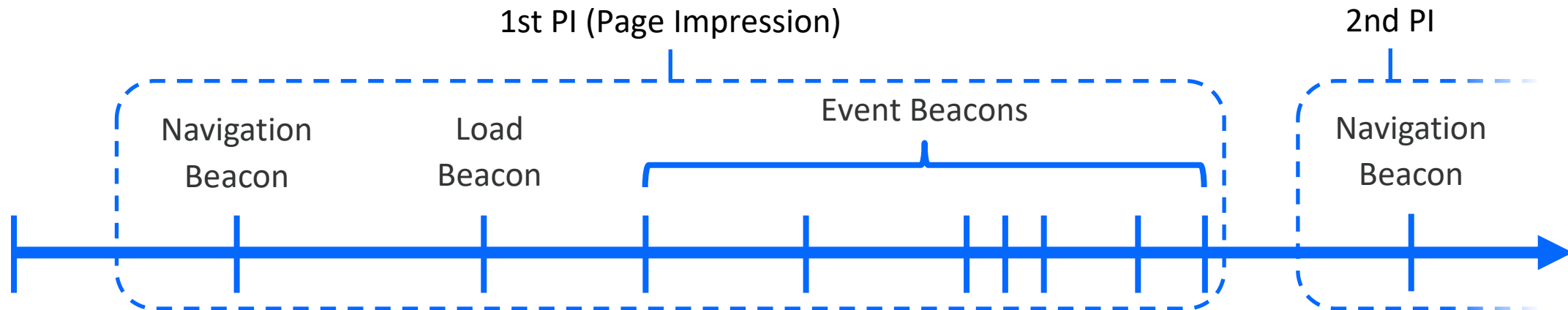
Running **Example**: Web Performance Analysis Pipeline





- **Goal:** verify that data in the pipeline is in an acceptable state for downstream processing, e.g.
 - External reporting (statistics, visualizations & dashboarding)
 - Internal reporting (debugging, product optimizations)
 - Decision-making (analytics, machine learning)
- **Data validation can be integrated in and between all stages**

Dimensions of Data Quality: **Completeness**



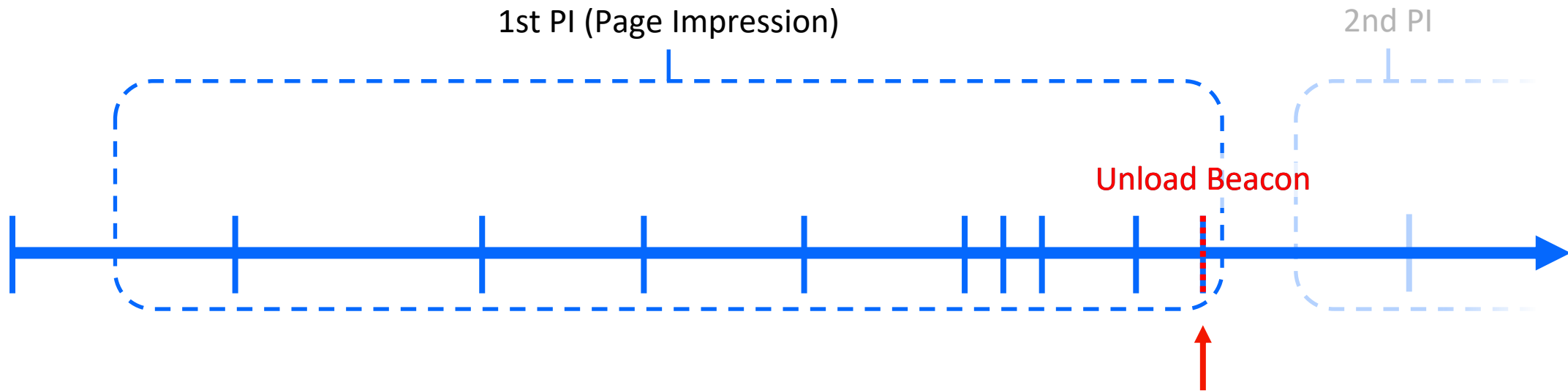
- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: **Completeness**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	"670ms"	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	0.256	456	abc.de/blue
13:29:17.468	faf55e	Edge	1.598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Unload Beacon **Reliability** as an Example Challenge



- Is beacon loss a problem at all?
- **When** is beacon loss a problem?
 - For which beacon types? For which beacon strategies?
- **Where** is beacon loss a problem?
 - For which browsers? For which device types?

Unload Beacon **Reliability** by Strategy

```
1 addEventListener("unload", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```

← available on all platforms

```
1 addEventListener("beforeunload", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```

```
1 addEventListener("pagehide", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```

```
1 addEventListener("visibilitychange", (event) => {  
2   if (document.visibilityState === 'hidden') {  
3     navigator.sendBeacon(url, JSON.stringify(data));  
4   }  
5 });
```

experimental feature

(only available as origin trial in Chrome)

```
1 var beacon = new window.PendingPostBeacon(  
2   url,  
3   {  
4     timeout: 60000,  
5     backgroundTimeout: 0  
6   });  
7 beacon.setData(JSON.stringify(data));
```



Unload Beacon **Reliability** by Strategy

```
1 addEventListener("unload", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```


← Available on all platforms

```
1 addEventListener("beforeunload", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```

```
1 addEventListener("pagehide", (event) => {  
2   navigator.sendBeacon(url, JSON.stringify(data));  
3 });
```

```
1 addEventListener("visibilitychange", (event) => {  
2   if (document.visibilityState === "hidden") {  
3     navigator.sendBeacon(url, JSON.stringify(data));  
4   }  
5 });
```




 Sophie Ferrlein. [Data Viz for Engineers: Optimizing Insight & Decision Making Through Visualization](#), code.talks (2023)

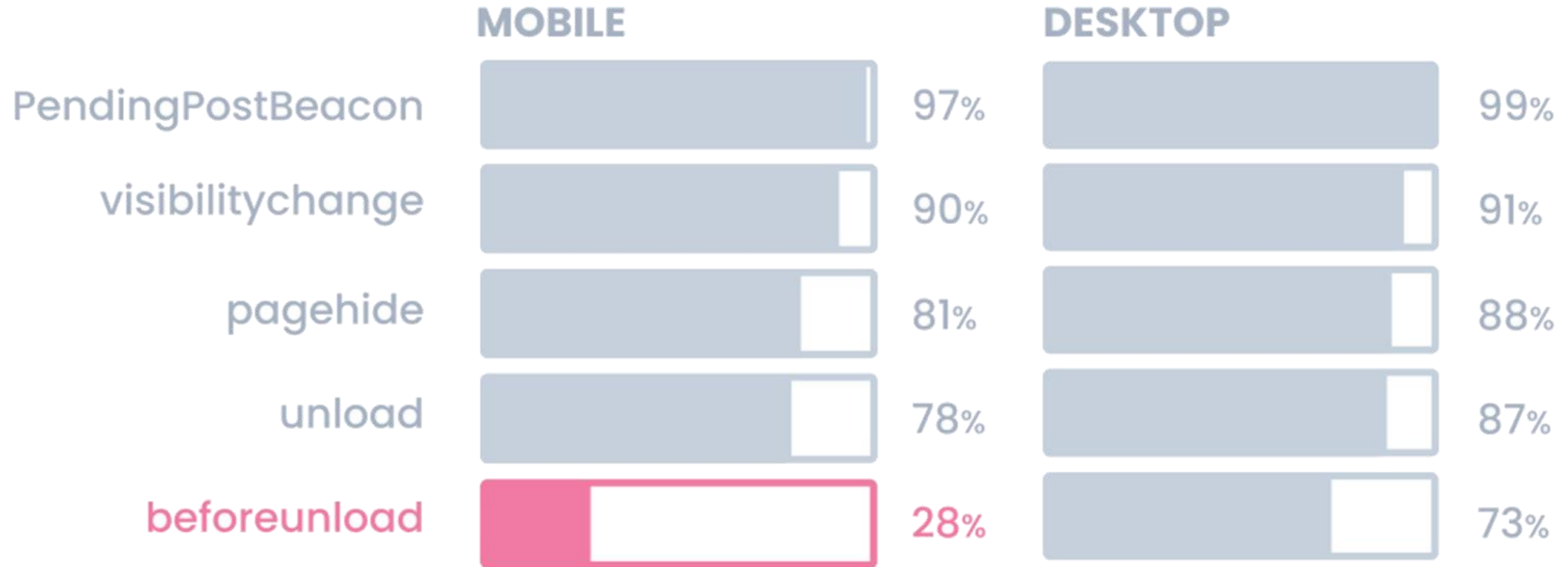
code.talks 2023, Sept. 14

 **Thursday, 14:00**
(Data Analytics & Science)


```
new window.PendingPostBeacon(  
  
60000,  
andTimeout: 0  
  
7 beacon.setData(JSON.stringify(data));
```

 Erik Witt. [Unload Beacon Reliability: Benchmarking Strategies for Minimal Data Loss](#), Speed Kit Tech Blog (2023)

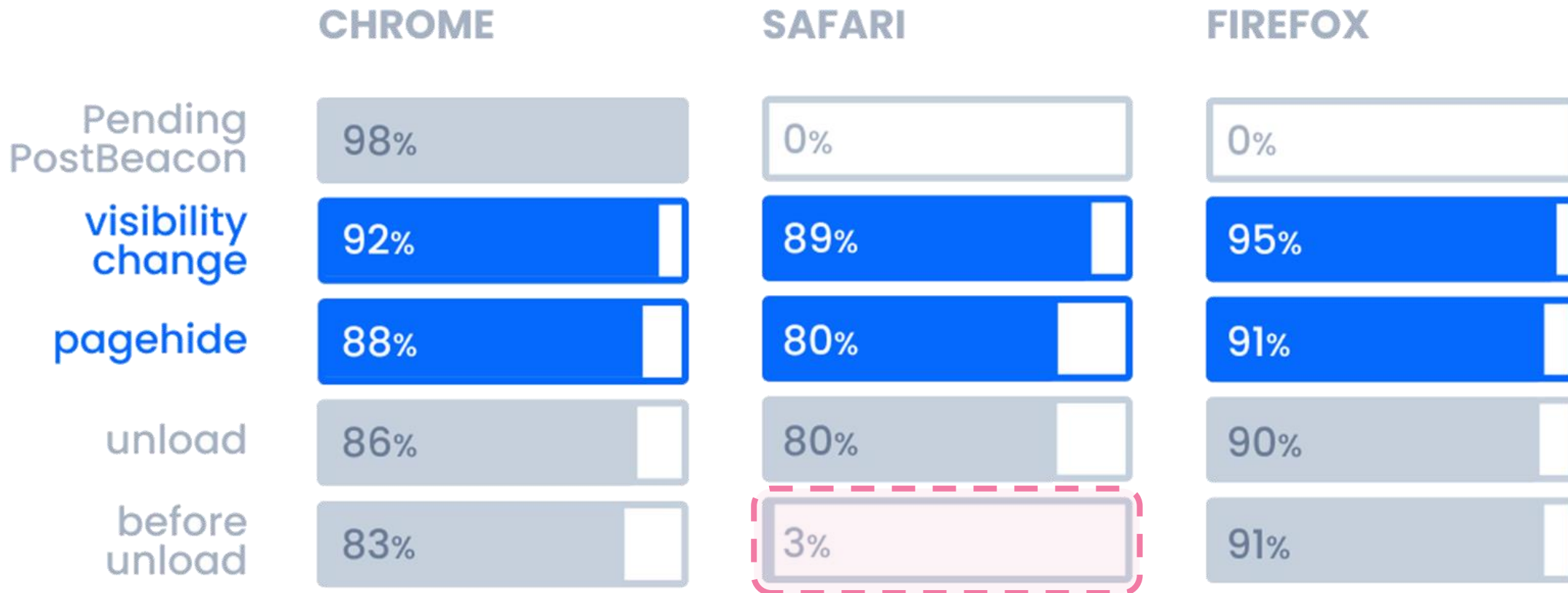
Unload Beacon **Reliability** by Strategy & Device



Based on 52 million page views on a globally operating e-commerce site measured by Speed Kits real user-monitoring | August 2023

 Desktop data may be overrepresented! 

Unload Beacon **Reliability** by Strategy & Browser



Based on 52 million page views on a globally operating e-commerce site measured by Speed Kits real user-monitoring | August 2023

Unload Beacon **Reliability**: The Ideal Combo Strategy

Can be used with
Backward-Forward Cache?



Based on 52 million page views on a globally operating e-commerce site measured by Speed Kits real user-monitoring | August 2023

Erik Witt. [Unload Beacon Reliability: Benchmarking Strategies for Minimal Data Loss](#), Speed Kit Tech Blog (2023)

Dimensions of Data Quality: **Consistency**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	"670ms"	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	0.256	456	abc.de/blue
13:29:17.468	faf55e	Edge	1.598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: Browser values should be unified for all records in the same session!

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: **Consistency**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	"670ms"	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	0.256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1.598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: Browser values should be unified for all records in the same session!
→ value may be replaced with majority vote (another reasonable option: replace old values with latest one)

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: **Consistency**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	"670ms"	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	0.256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1.598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: All values represent milliseconds, but formats differ depending on browser.

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - **Consistency: Does data have a valid format and does it comply with business semantics?**
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: **Consistency**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	670	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: All values represent milliseconds, but formats differ depending on browser.
→ values may be converted to integer

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - **Consistency: Does data have a valid format and does it comply with business semantics?**
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: Accuracy

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	670	123	null
13:26:48.139	9cddf7	Firefox	692 654	456	abc.de/red
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: Despite being in the right format, one value does not represent a reasonable timer value.

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - **Accuracy: Do data items represent their corresponding real-world entities well?**
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

Dimensions of Data Quality: Accuracy

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	670	123	null
13:26:48.139	9cddf7	Firefox	null	456	abc.de/red
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: Despite being in the right format, one value does not represent a reasonable timer value.

→ broken value may be removed

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - **Accuracy: Do data items represent their corresponding real-world entities well?**
 - *Uniqueness: Are duplicate records known and are all unique attributes actually distinct?*

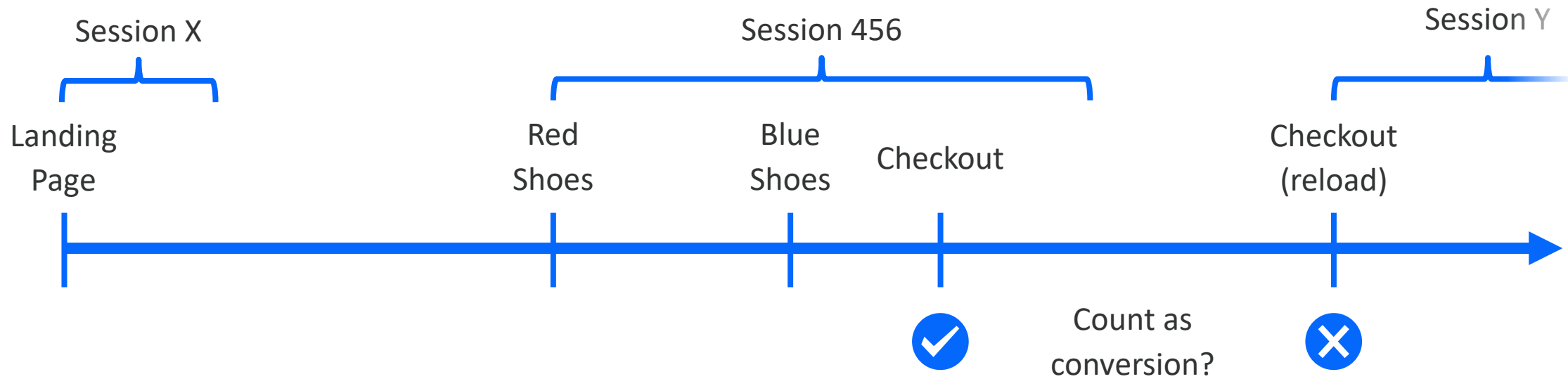
Dimensions of Data Quality: **Uniqueness**

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	670	123	null
13:26:48.139	9cddf7	Firefox	null	456	abc.de/red
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: The ID field should be unique, but two different records share the same value!

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - **Uniqueness: Are duplicate records known and are all unique attributes actually distinct?**

Dimensions of Data Quality: **Uniqueness**



- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - **Uniqueness: Are duplicate records known and are all unique attributes actually distinct?**

Dimensions of Data Quality: **Uniqueness**

Timestamp.	Pageload ID	Browser	LCP (Performance)	Session ID	URL
09:05:04.578	37ab08	Edge	670	123	null
13:26:48.139	9cddf7	Firefox	null	456	abc.de/red
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold
20:45:38.941	faf55e	null	null	null	abc.de/sold

Note: The ID field should be unique, but two different records share the same value!
→ merge duplicates into a single record

- There are different **dimensions of data quality**, especially:
 - *Completeness: Do we have all the data we need to assess page load performance?*
 - *Consistency: Does data have a valid format and does it comply with business semantics?*
 - *Accuracy: Do data items represent their corresponding real-world entities well?*
 - **Uniqueness: Are duplicate records known and are all unique attributes actually distinct?**

Maintaining Data Quality With **Constraints**

- **Constraints** are rules, conditions, or limits that data must adhere to
- **Type Checks** represent expectations on the data format, e.g.
 - *Value range* for numerical data (e.g. [0,MAX_INTEGER) for load timers)
 - *Format or pattern* for string-valued data (e.g. ISO 8601 for timestamps)
 - *Structure* for complex attributes (e.g. required keys for JSON objects)
- **Complex conditions** can further describe complex semantics such as
 - Cross-field or cross-record relationships (e.g. same browser within sessions)
 - Referential integrity between records in different collections
 - Custom constraints for domain semantics

Example: Declarative Constraints With Pandera (1/3)

```
import pandas as pd
import pandera as pa
from pandera import Column, DataFrameSchema, Check

# Define the schema
schema = DataFrameSchema(
    {
        "Timestamp": Column(pa.DateTime),
        "PageLoad ID": Column(pa.String,
            Check(lambda x: x.str.len() == 8)),
        "Browser": Column(pa.String,
            Check(lambda x: x.isin(["Chrome", "Edge", "Firefox"]))),
        "LCP": Column(pa.Int,
            Check(lambda x: (x >= 0) & (x <= 600000))),
        "Session ID": Column(pa.Int),
    }
)
```

Example: Declarative Constraints With Pandera (2/3)

```
# valid data item
good = {
    "Timestamp": [pd.Timestamp("2023-05-10 13:26:48.139")],
    "PageLoad ID": ["9cddf7"],
    "Browser": ["Firefox"],
    "LCP": [256],
    "Session ID": [456],
}

# invalid data item
bad = {
    "Timestamp": [pd.Timestamp("2023-05-10 09:05:04.578")],
    "PageLoad ID": ["37ab08"],
    "Browser": ["Edge"],
    "LCP": [692654], # timer value out of bounds
    "Session ID": [123],
}
```

Example: Declarative Constraints With Pandera (3/3)

```
for record in [good, bad]:
    record_id = record['PageLoad ID'][0]
    try:
        validated = schema(pd.DataFrame(record))
        print(f"\nValidation passed for record {record_id}!")
    except pa.errors.SchemaError as e:
        print(f"\nValidation FAILED for record {record_id}:")
        print(e)
```

```
Validation passed for record 9cddf7!
```

```
Validation FAILED for record 37ab08
```

```
<Schema Column(name=LCP, type=DataType(int64))> failed element-wise  
validator 0:
```

```
<Check <lambda>>
```

```
failure cases:
```

index	failure_case
0	0 692654

Fundamental Challenge: Scalability

One Month in Data Errors at Baqend: April 2023

Data Errors by Type

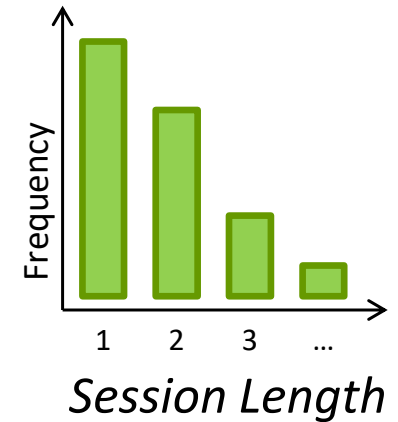


Data Errors by Cause



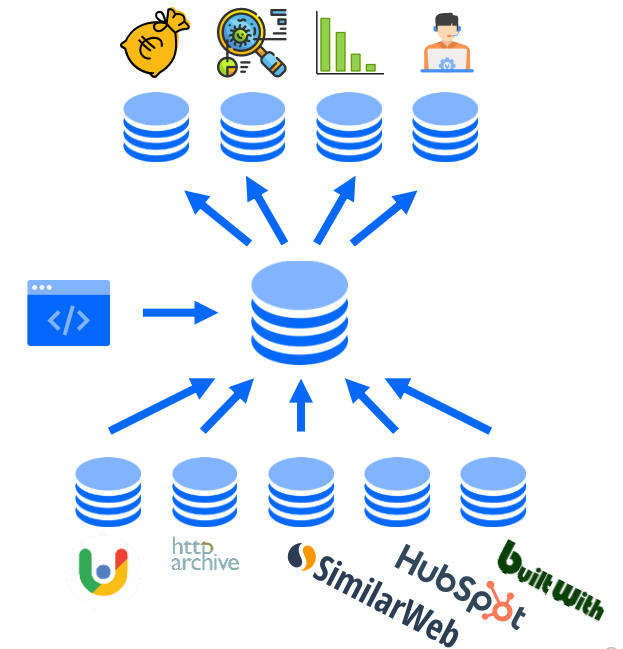
Challenging at Scale: Complexity

Timestamp	Pageload ID	Browser	LCP (Performance)	Session ID	URL	...	Session Length	Conversion
09:05:04.578	37ab08	Edge	670	123	null	...	1	0
13:26:48.139	9cddf7	Firefox	null	456	abc.de/red	...	3	1
13:28:23.857	0b577a	Firefox	256	456	abc.de/blue	...		
13:29:17.468	faf55e	Firefox	1598	456	abc.de/sold	...		
20:45:38.941	faf55e	null	null	null	abc.de/sold	...		

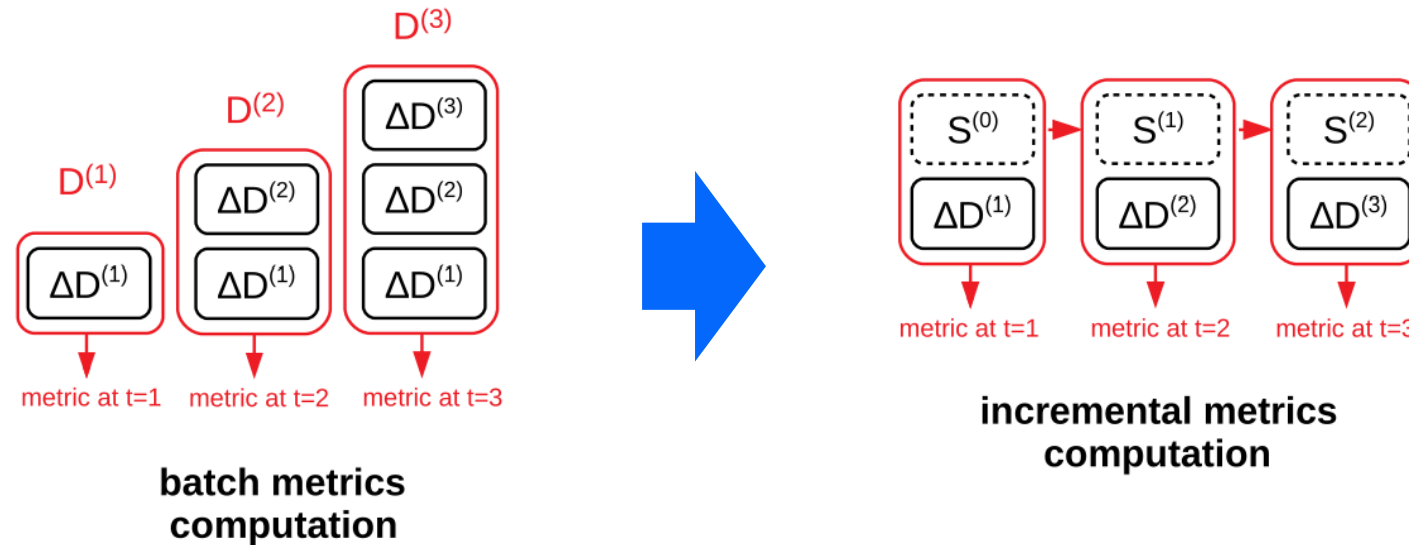


- Manual constraint definition is often infeasible, because of ...
 - ... inherent data complexity (often hundreds of attributes)
 - ... aggregation, derived storage, and evolving schemas
 - ... a plethora of other data stores to integrate!

→ **Automation** is necessary!



Challenging at Scale: **Continuity**



- Computing validation metrics from scratch periodically can be infeasible, because of ...
 - ... strict timing requirements
 - ... efficiency or cost reasons
 - ... data privacy reasons

Details in Felix' talk!

→ **Incremental computation** can be the only option!

W. Wingerath, B. Wollmer, M. Bestehorn, S. Succo, F. Bücklers, J. Domnik, F. Panse, E. Witt, A. Sener, F. Gessert, N. Ritter. Beaconnect: Continuous Web Performance A/B-Testing at Scale, VLDB (2022)

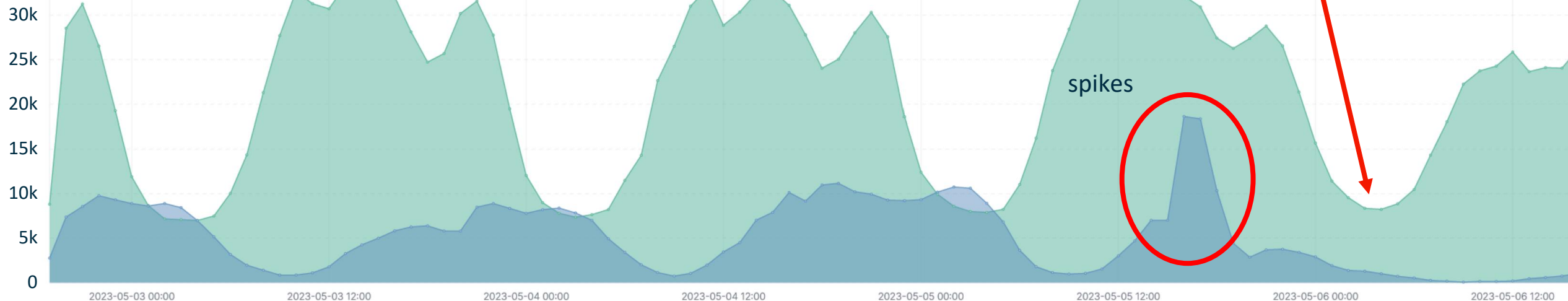
Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann: Automating Large-Scale Data Quality Verification, VLDB 2018.

Challenging at Scale: **Volatility**

● EU Customer
● US Customer

geo-distributed
customer base

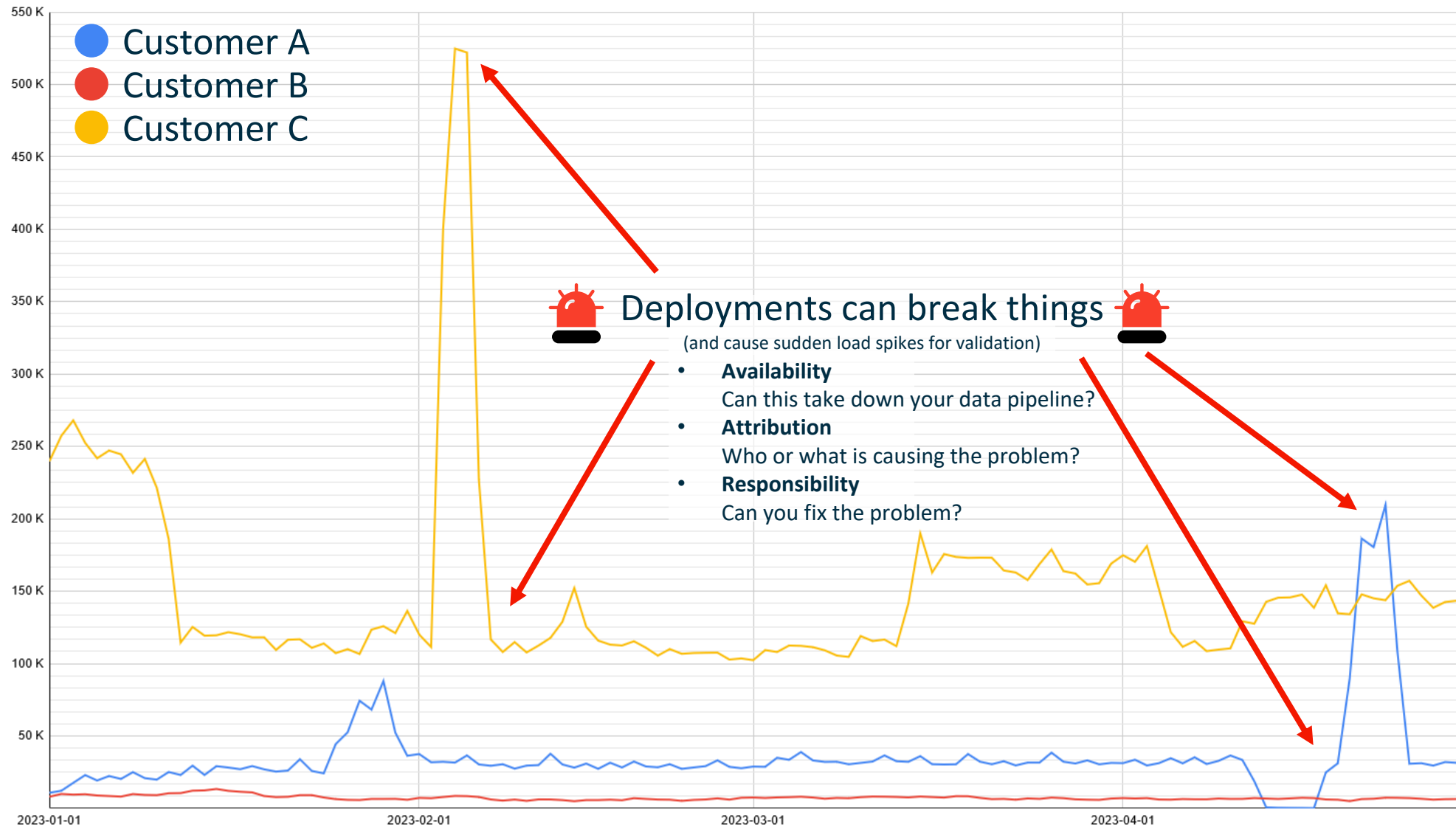
Data Errors Per Hour
(May 03-06, 2023)



- Specifying generalized constraints can be difficult in large deployments, because of ...
 - ... temporal fluctuations (e.g. throughout the day, on black Friday, or during holidays)
 - ... multi-tenancy (e.g. different data patterns by customer timezone or domain)

→ **Elasticity & Multi-Tenancy** requirements can be challenging!

Challenging at Scale: „Continuity“ (Continuity + Volatility)



So How Do You **Handle** All This?

- **Advanced Techniques**

- Inferring constraints
- Adapting to schema changes
- Incremental computation of complex measures

- **Tooling & Frameworks**

- Validation libraries such as Great Expectations, Pandera, TFDV, or Deequ
- Preprocessing and validation with Apache Spark and Apache Flink

- **Further Challenges**

- Handling distribution (validation per partitioning, avoiding skew, ...)
- Efficiency and performance (load distribution, approximation, ...)
- Operational challenges (anomaly detection, fixing, load shedding, ...)



Data Validation at Scale: **Summary**

- **Data Quality** can be measured along dimensions such as completeness, consistency, accuracy, and uniqueness
- **Constraints** specify expectations about the data and can be used to enforce them
- **Data Validation** is the process of ensuring high data quality for processes like analysis, modeling, and decision-making
- Data Validation **Challenges at Scale** include
 - *Complexity*: schemas are often too complex to define constraints manually
 - *Volatility*: data varies throughout the day, by season, or by customer
 - *Continuity*: incremental processing is required when computation from scratch is infeasible



Thanks! **Questions** ?



Material Available at <https://wolle.science>