Time Series Analysis

… using an Event Streaming Platform

Mirko Kämpf - SA @ Confluent, Inc. - *Team CEMEA*
Meetup - Leipzig

Time Series Analysis

… using an Event Streaming Platform

Advanced time series analysis (TSA) requires very special data preparation procedures to convert raw data into compatible formats.

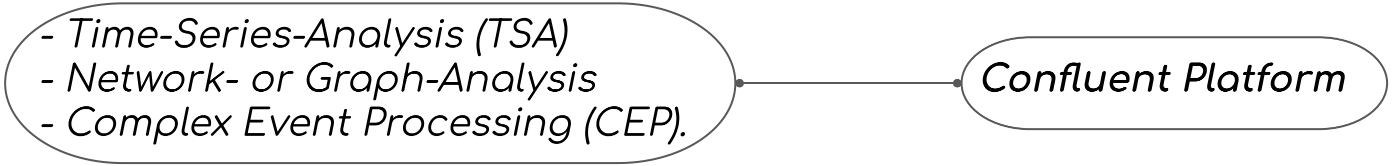
In this presentation you will see typical processing patterns for TSA, from simple statistics to reconstruction of correlation networks and interaction graphs.

The first case is relevant for anomaly detection and to protect safety.

Reconstruction of graphs from time series data is a very useful technique to better understand complex systems like supply chains, material flows in factories, information flows within organizations, and especially in medical research.

With this motivation we will look at typical data aggregation patterns, how to apply analysis algorithms in the cloud, and into a reference architecture for TSA on top of the Confluent Platform, which is baked by Apache Kafka.

This presentation is about linking:



- Time-Series-Analysis (TSA)
- Network- or Graph-Analysis
- Complex Event Processing (CEP).

Confluent Platform

Research work ends often with nice charts, scientific papers, and conference talks.

But, many published results **can't** be reproduced -
often because the setup it is simply too complicated …

Question:
How can we integrate data streams,
experiments, and decision making better?

Why not using batch processing?

Study the anatomy ...

- **Batch processing is fine:**
  - as long as your data doesn’t change.
  - in PoCs for method Development in the Lab.
  - For research in fixed scope.

Why using Kafka?

- **Stream processing is better:**
  - for real time business in changing environments.
  - iterative (research) projects.
  - repeatable experiments on replayed data.

Study and influence
the living system ...

Let’s **stream the title :**

From Events to Time Series …

to Graphs … to Events …

for better Decisions

WHAT?

WHY?

Content:

(1) Intro

Typical types of event

How to identify hidden events?

3 aspects around advanced analytics:

- Complex event analysis
- Integration across domains
- Extraction of hidden events

(2) The Challenge

(3) Approach

Time Series Analytics &
Network Analytics
in Kafka

Create time series from events

Create graphs from time series pairs

(4) Architecture:

Simplified architecture for CSA

Reusable building blocks for CSA

Events - 1

Business events

- transaction records
- discrete observation

How to handle events?

JUTS SIMPLE
OBSERVATION &
DATA CAPTURING

A Sale

An Invoice

A Trade

A Customer Experience

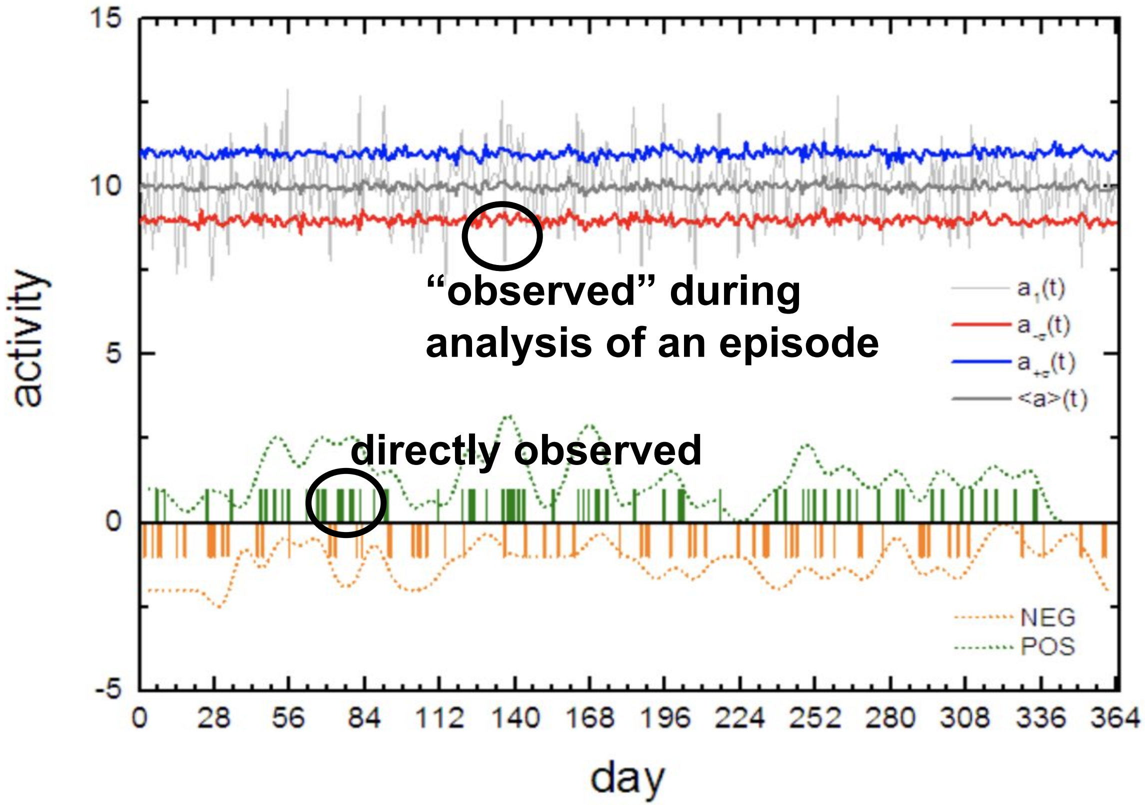
Events - 2

Well defined events

- in known context

How to identify events?

Sometimes: SIMPLE
Sometimes: DATA ANALYSIS



“observed” during analysis of an episode

directly observed

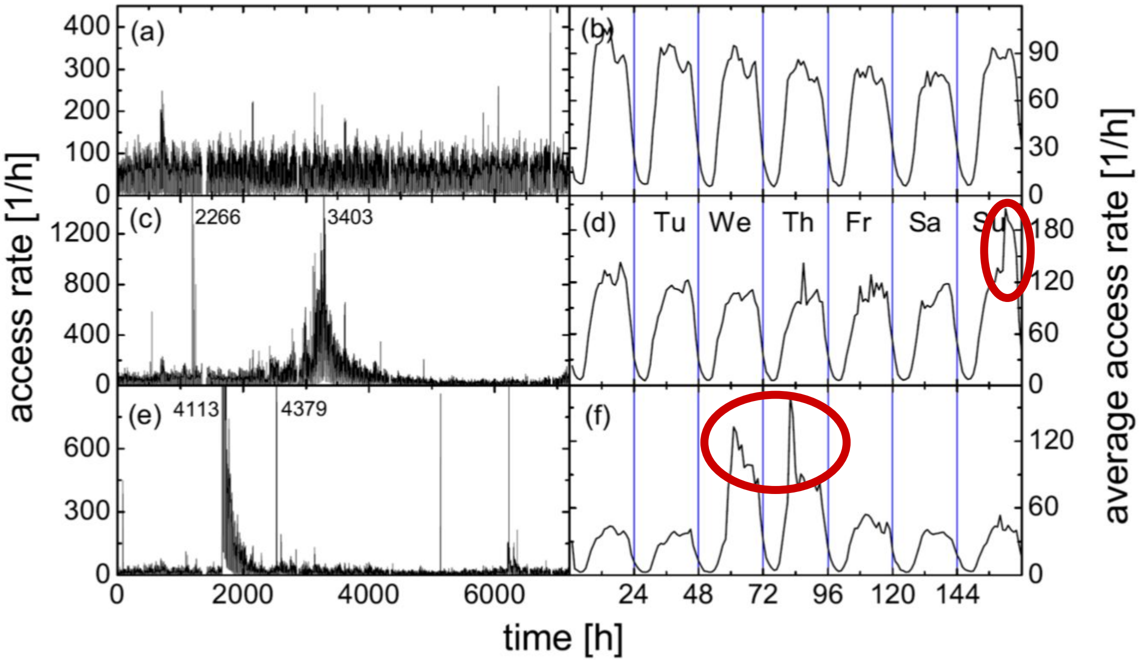
Events - 3

Extreme Events

- *“outliers”* in unknown context

How to handle?

ADVANCED
DATA ANALYSIS *(& ML)*



Reality is Complex:

We should simplify a bit!

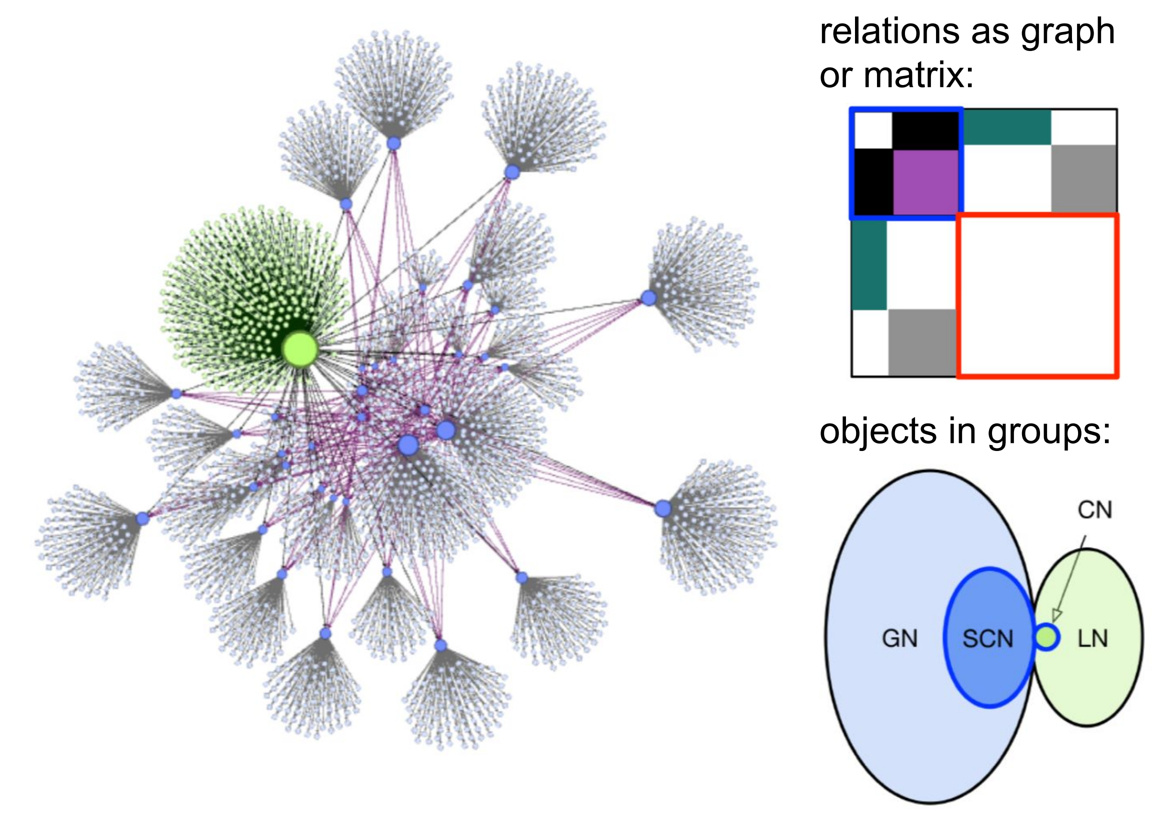
Simplification in our method can lead to isolation:

- DATA SILOS
- OPERATIONAL SILOS

SOLUTION:

GRAPHS capture structure.

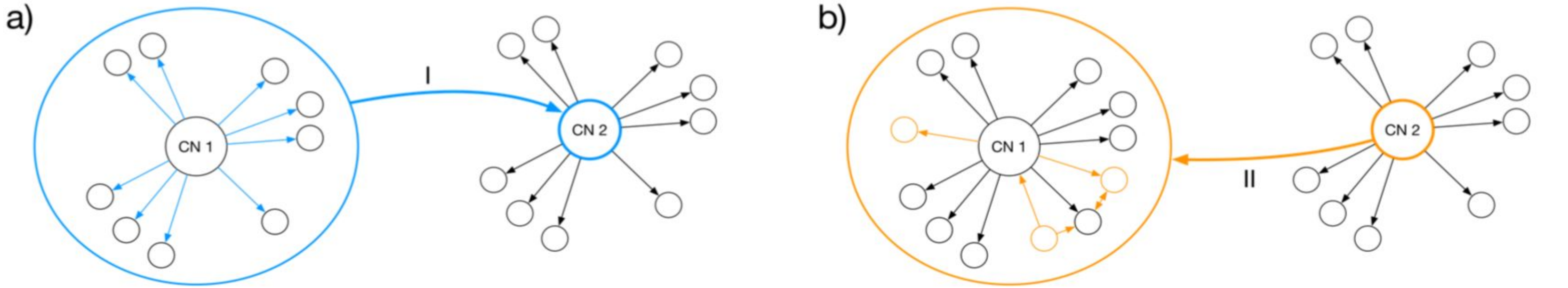
TIME SERIES capture properties over time (history).



relations as graph or matrix:

objects in groups:

Interacting Subsystems ⇒ Multi-Layer-Networks



Layer 1: *Neighborhood structure of CN 1 has impact on a node CN 2.*

Layer 2: *Node CN 2 has impact on neighborhood structure of CN 1.*

Dynamic processes can cause inter-dependencies between layers.

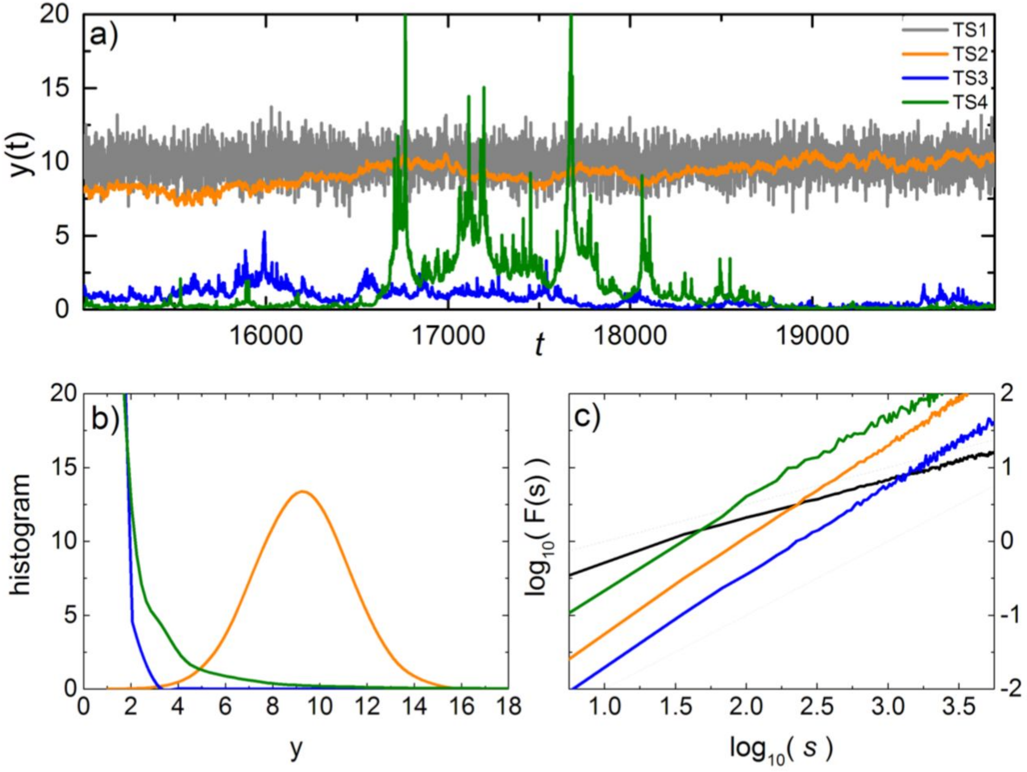
Such dependencies cause effects which are not measurable directly.
>>> This is the reason for using the methodology!!!

Univariate TSA: single episodes are processed

- Distribution of values

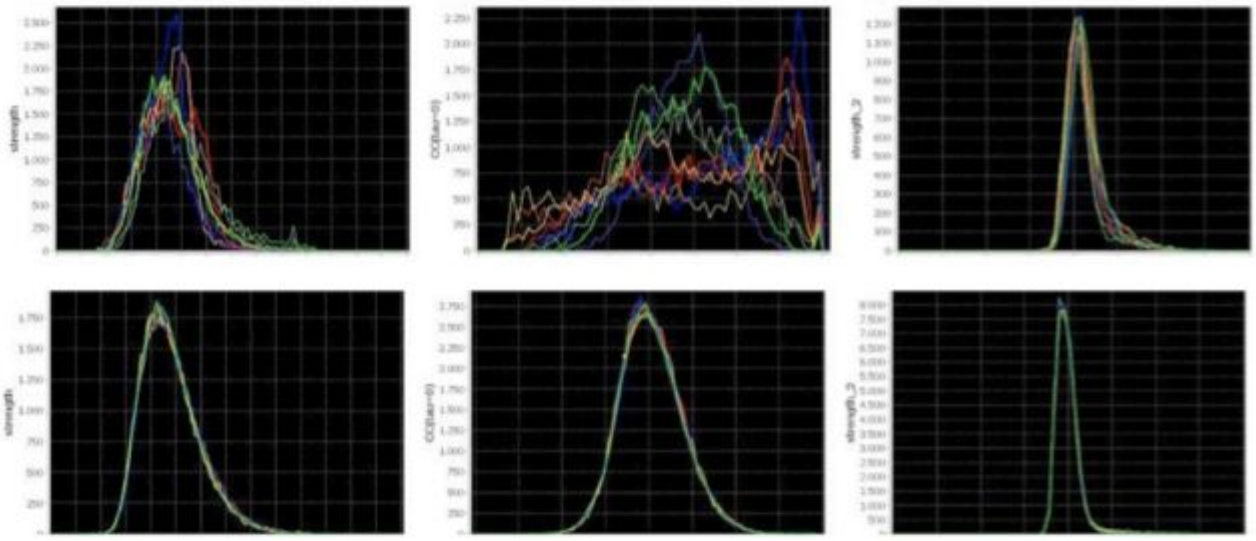
- Fluctuation properties

- Long-term correlations (memory effects)

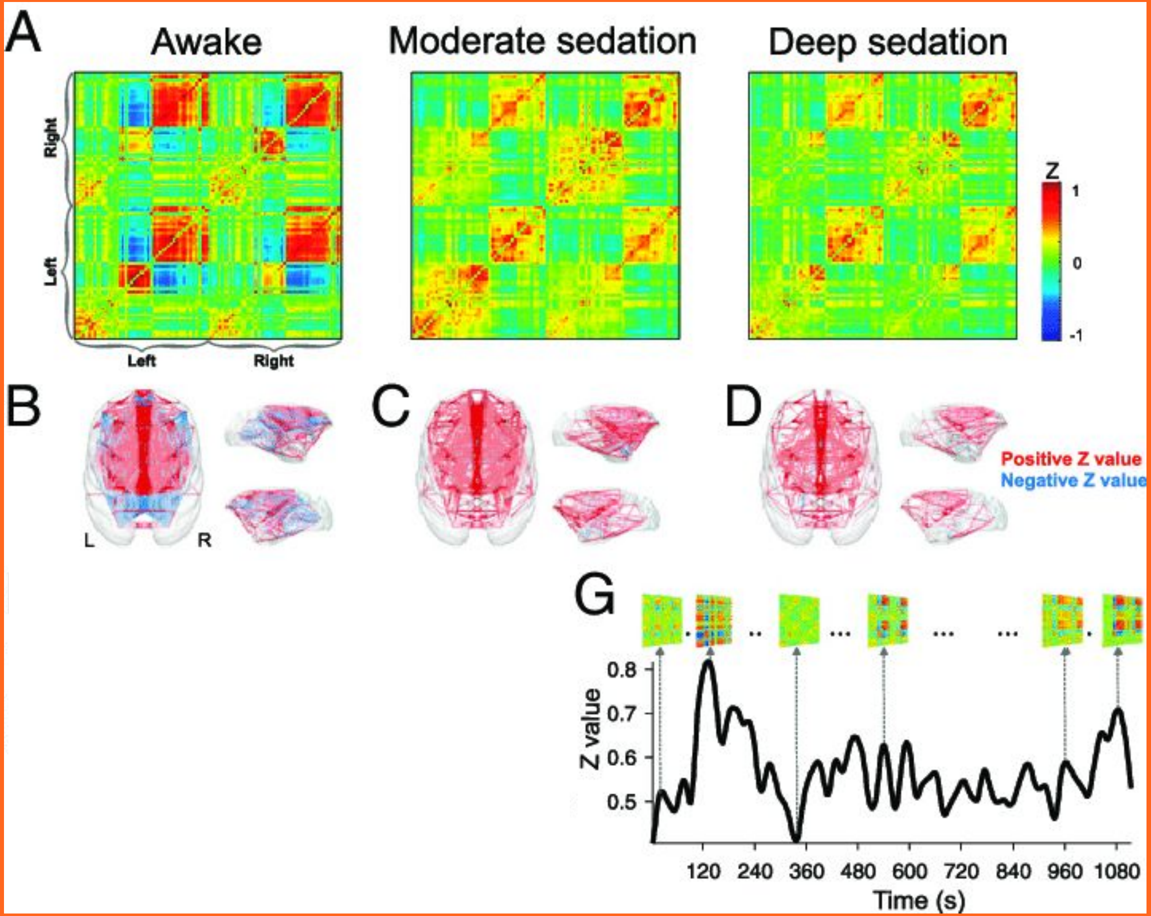


Multivariate TSA: pairs / tuples of episodes are processed

- Comparison Similarity measures for link creation



Distribution of cross-correlation coefficients for pairs of access-rate time series of Wikipedia pages (top) compared to surrogat data (bottom) - 100 shuffled configurations are considered


A
Awake
Moderate sedation
Deep sedation
Right
Left
Left
Right
Z
1
0
-1
B
L
R
C
D
Positive Z value
Negative Z value
G
Z value
0.8
0.7
0.6
0.5
120
240
360
480
600
720
840
960
1080
Time (s)

WHAT?

WHY?

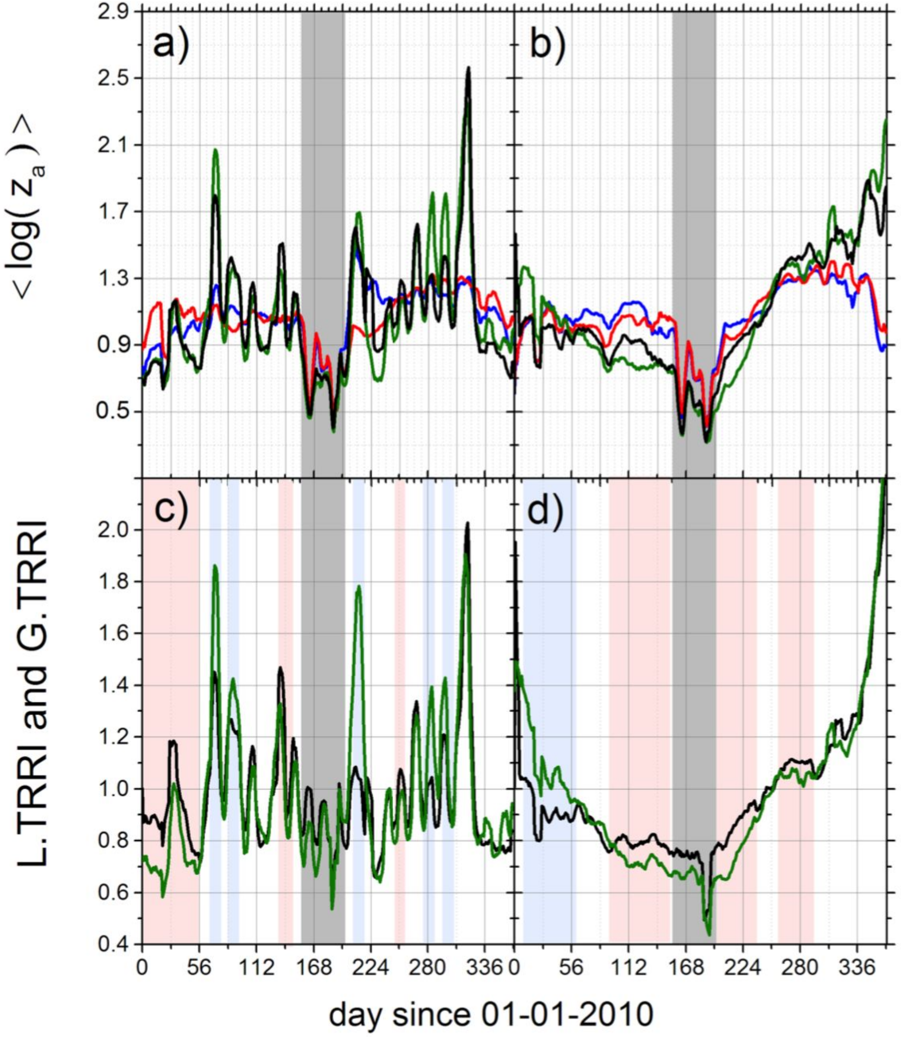
Events - 4

Hidden Events

- invisible state changes in complex systems

How to handle?

Contextual
TIME SERIES ANALYSIS &
NETWORK Topology ANALYSIS



confluent

Recap:

What events are and how to process event-data is often misunderstood *or simply unclear.*

It all depends on our view and our goals!

IT Operations

- Server crash
- Cyber crime

Special procedures established or under construction

Business Events

- Big deal won
- Technical issue solved

The events which make people & the market happy :-)

Transactions (in business)

- orders placed
- products shipped
- bills paid

Event Driven Architecture

Extreme Events:

- Service slow down due to emerging bottlenecks
- Increased demand in a resource

Complex Event Analysis

Things become complicated:

Complex Event Analysis
Integration Across Domains
Extraction of Hidden Event

METHODOLOGY

Complex Event Analysis

- time series analysis and ML reveal hidden events
- multi-stage processing is usually needed

ORGANIZATION & OPERATIONS

Integration Across Domains

- distributed event processing systems are used
- apps consume and produce events of different flavors
- Event-types and data structures my change over

confluent

TECHNOLOGY & SCIENCE

Extraction of Hidden Events

- requires **Applied** Data **Analysis** & Data **Science**
- embedding of **Complex Algorithms** in IT landscape
- integration of **GPU/HPC** and data pipelines

The Challenge:

How can we combine unbound data assets and scientific methods?

A. you pipe the data to the place where it can be processed easily, e.g., to the cloud or into special purpose systems.

B. you integrate complex algorithms in your processing pipeline.

Problems on ORGANIZATION level:

Legacy systems in the fab can’t be integrated without additional expensive servers. Often, this data is unreachable.

Business data is managed by different teams using different technologies.

Data scientists play with some data in the cloud, and they all do really L❤️VE notebooks. But often, they don’t know CI/CD systems.

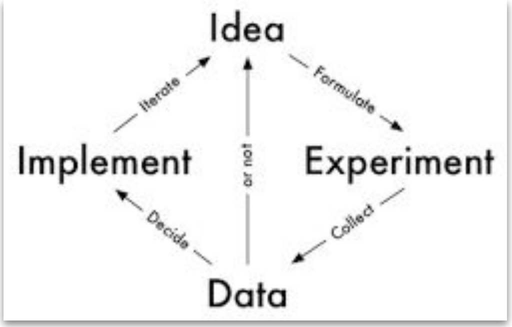
Kafka and its Ecosystem …

- are considered to be middleware, managed by IT people:
  - researchers do not plan their experiments around such a technology.
- don’t offer ML / AI components:
  - many people think, that a model has to be executed on the edge-device or in the cloud.

Just because they don’t understand doesn’t mean you’re on the wrong path.

@QWORLDSTAR

Kafka can support agile experiments …



- it gives access to data (flows) in real time,
  - in a way, which allows a replay of experiments at a later point in time
  - completely managed by Confluent

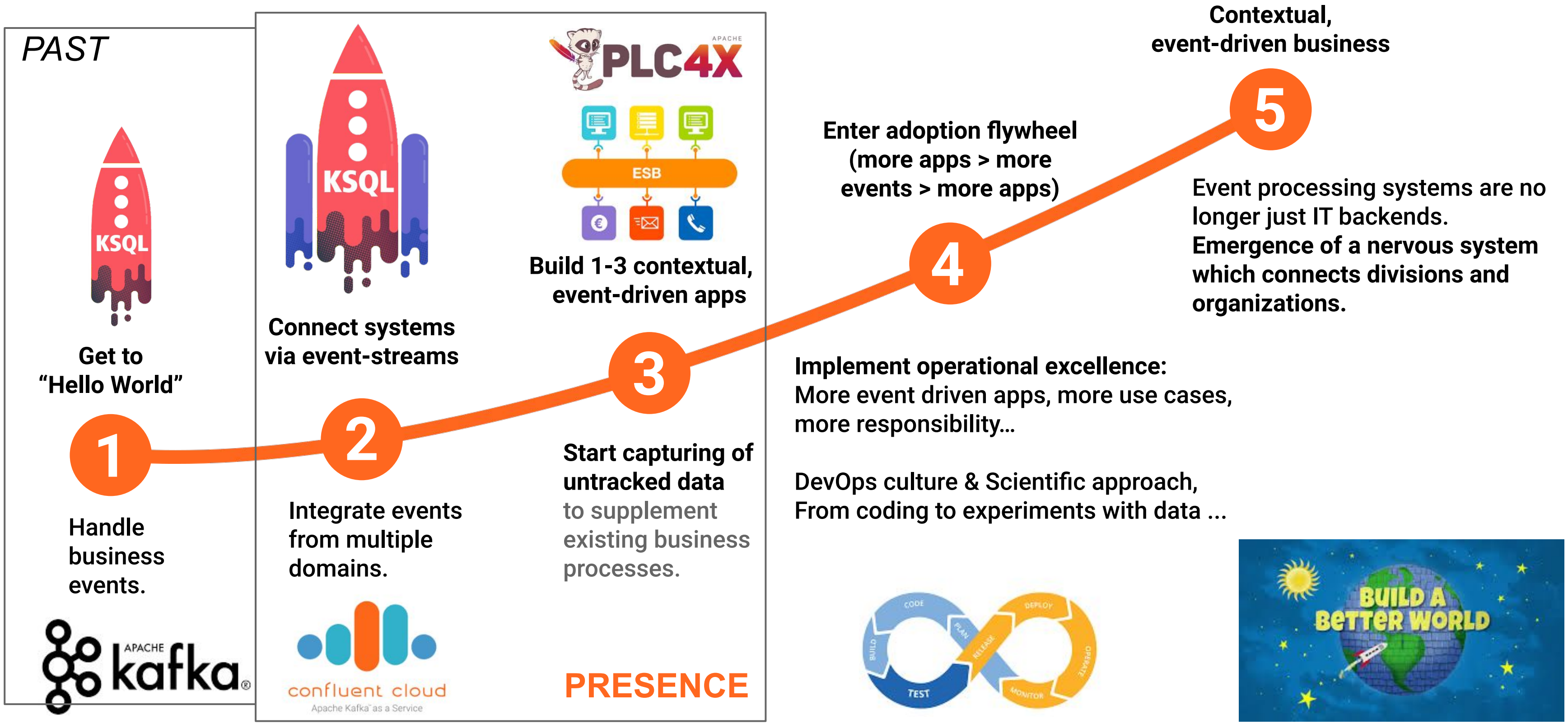
- allows **variation of analysis without redoing the same experiment** by simply reusing the persisted event-stream again.

- Kafka Streams and KSQL allow data processing in place
  - this allows faster iterations because plausibility checks can be done in place
  - the streaming API gives freedom for extension
  - DSL and KSQL save you a lot of time

Why not building on top of the right tools ???

How to make use of a variety of event data:

for an Event-Driven Business / Research?



PAST

Get to “Hello World”

1

Handle business events.

Connect systems via event-streams

2

Integrate events from multiple domains.

Build 1-3 contextual, event-driven apps

3

Start capturing of untracked data to supplement existing business processes.

PRESENCE

Enter adoption flywheel (more apps > more events > more apps)

4

Implement operational excellence:
More event driven apps, more use cases, more responsibility…

DevOps culture & Scientific approach,
From coding to experiments with data ...

Contextual, event-driven business

5

Event processing systems are no longer just IT backends.
Emergence of a nervous system which connects divisions and organizations.

ADVANCED
TIME SERIES ANALYSIS &
NETWORK ANALYSIS

... and how both fit into Kafka.

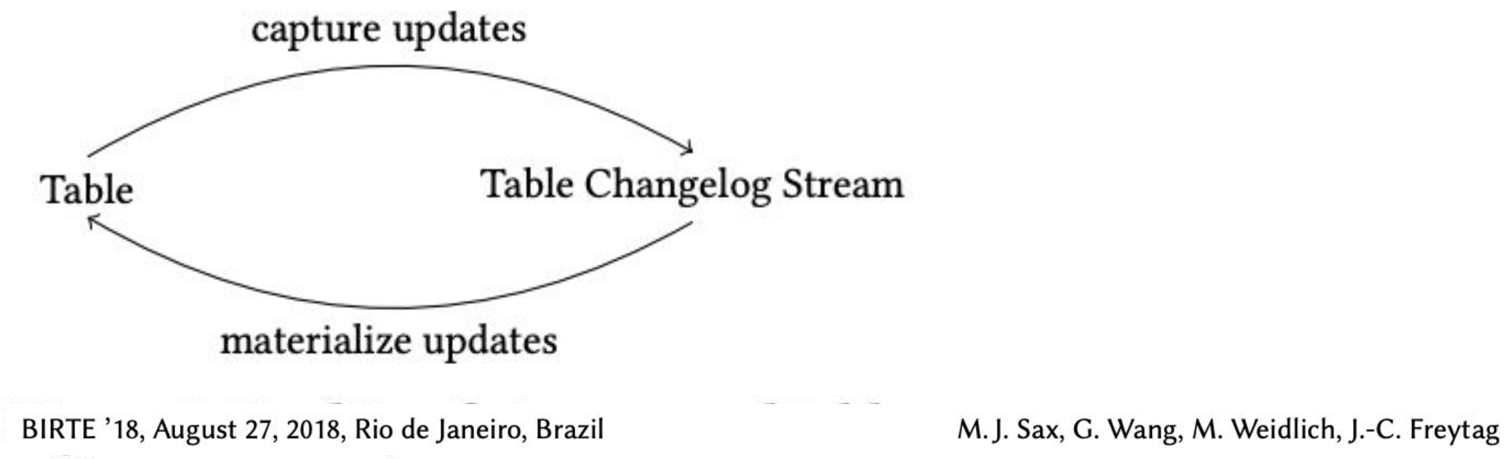
METHODOLOGICAL aspects:

How do time series analysis and graphs fit into Kafka’s data model?

I think, Kafka is a messaging system? Or am I wrong?

Please, tell me, how can I use Kafka for advanced analytics
or even machine learning?

Table Stream Duality



BIRTE ’18, August 27, 2018, Rio de Janeiro, Brazil

M. J. Sax, G. Wang, M. Weidlich, J.-C. Freytag

Table - Stream Duality

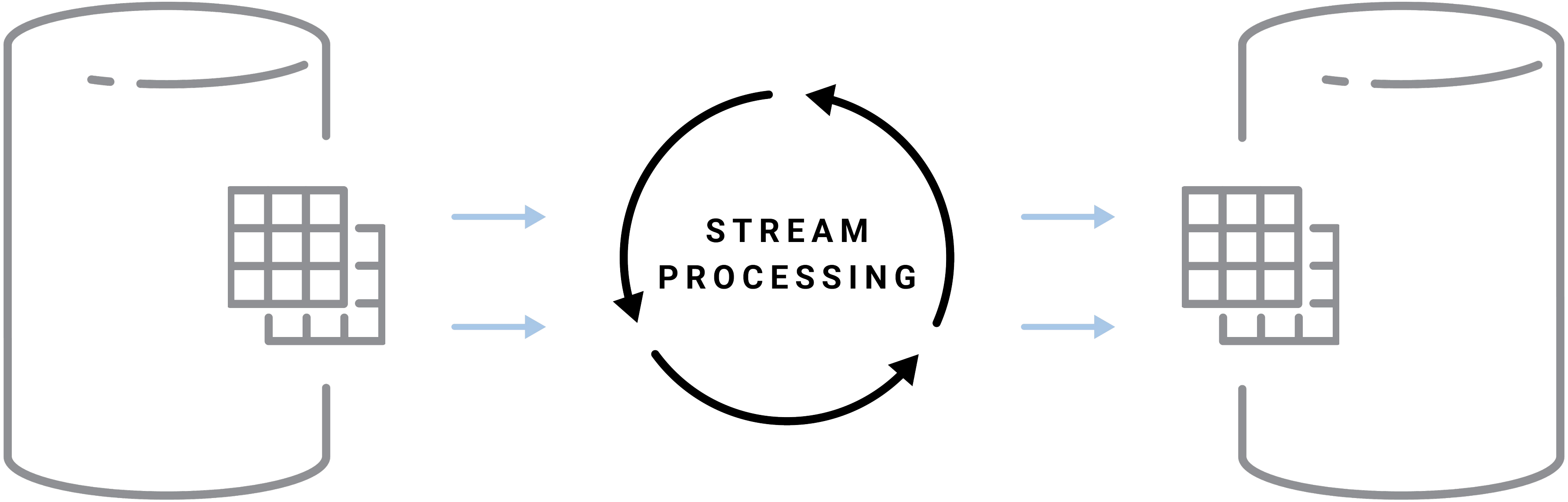
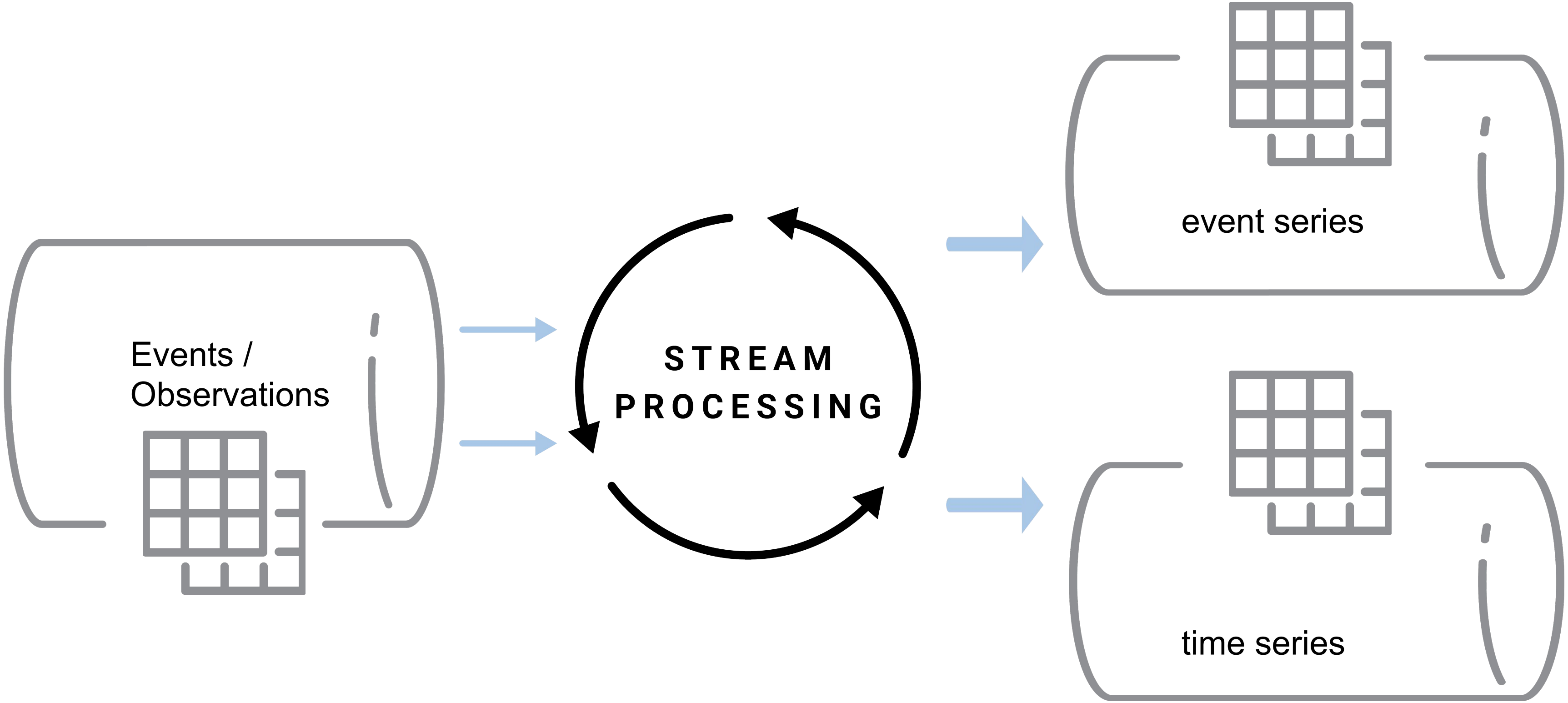


Table Stream Duality ⇒ Time Series and Graphs

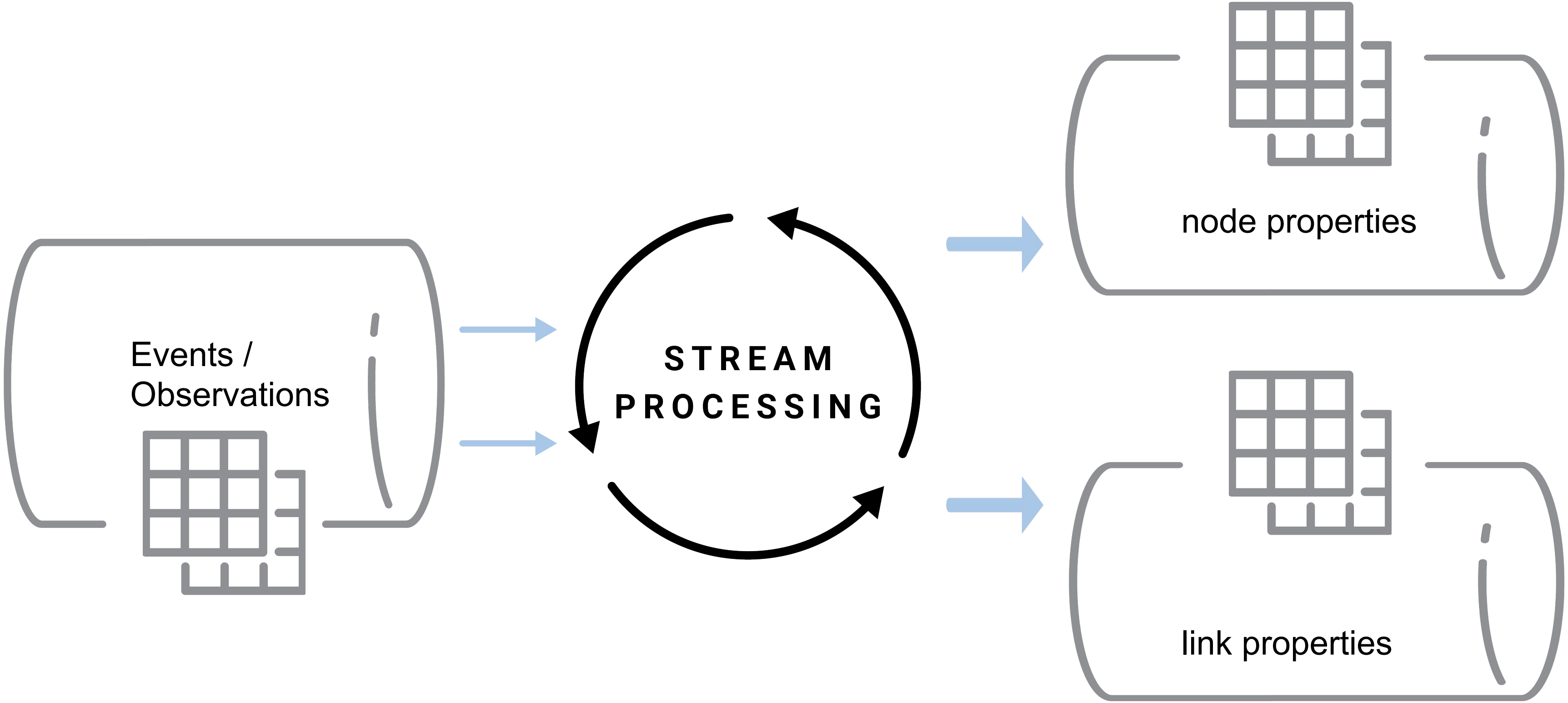
A ***time series*** is a table of **ordered observations** in a fixed context.

A ***graph*** can be seen as a list of nodes and a list of links - properties are stored in **two tables**.

Create Time Series from Event Streams:
By Aggregation, Grouping, and Sorting



Create Networks from Event Streams:
By Aggregation, Grouping, and Sorting



From Table of Events to - Time Series

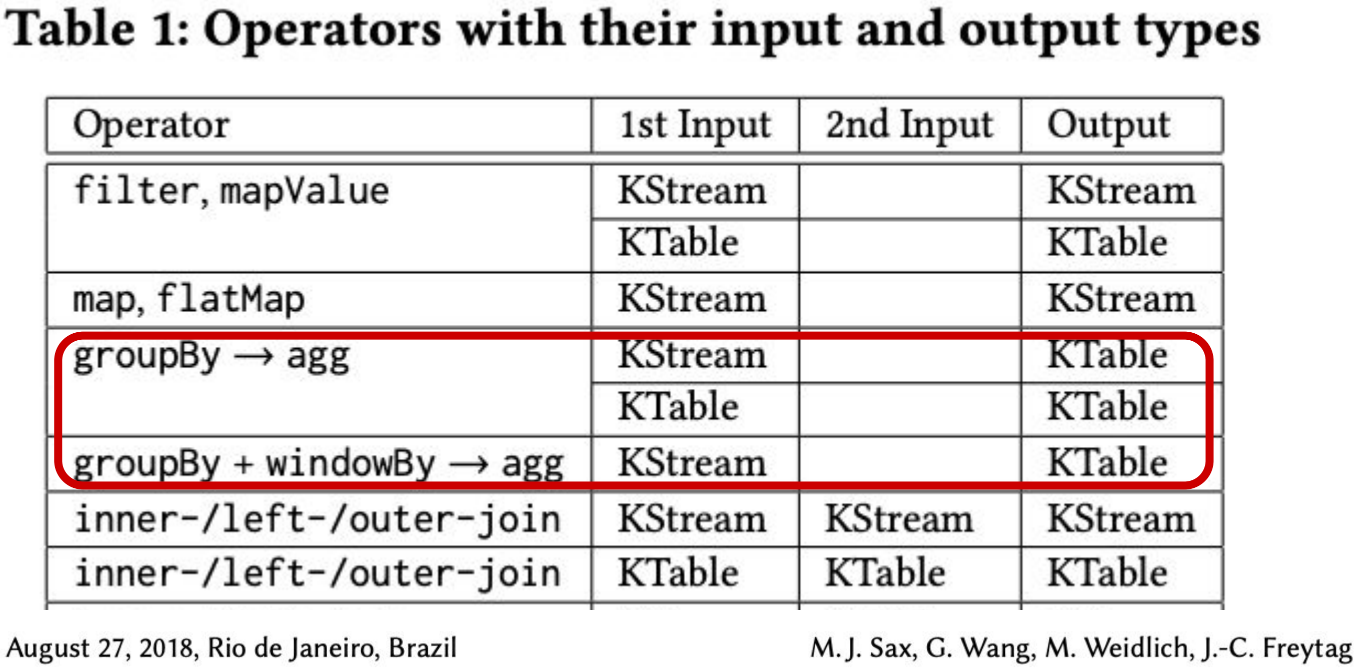
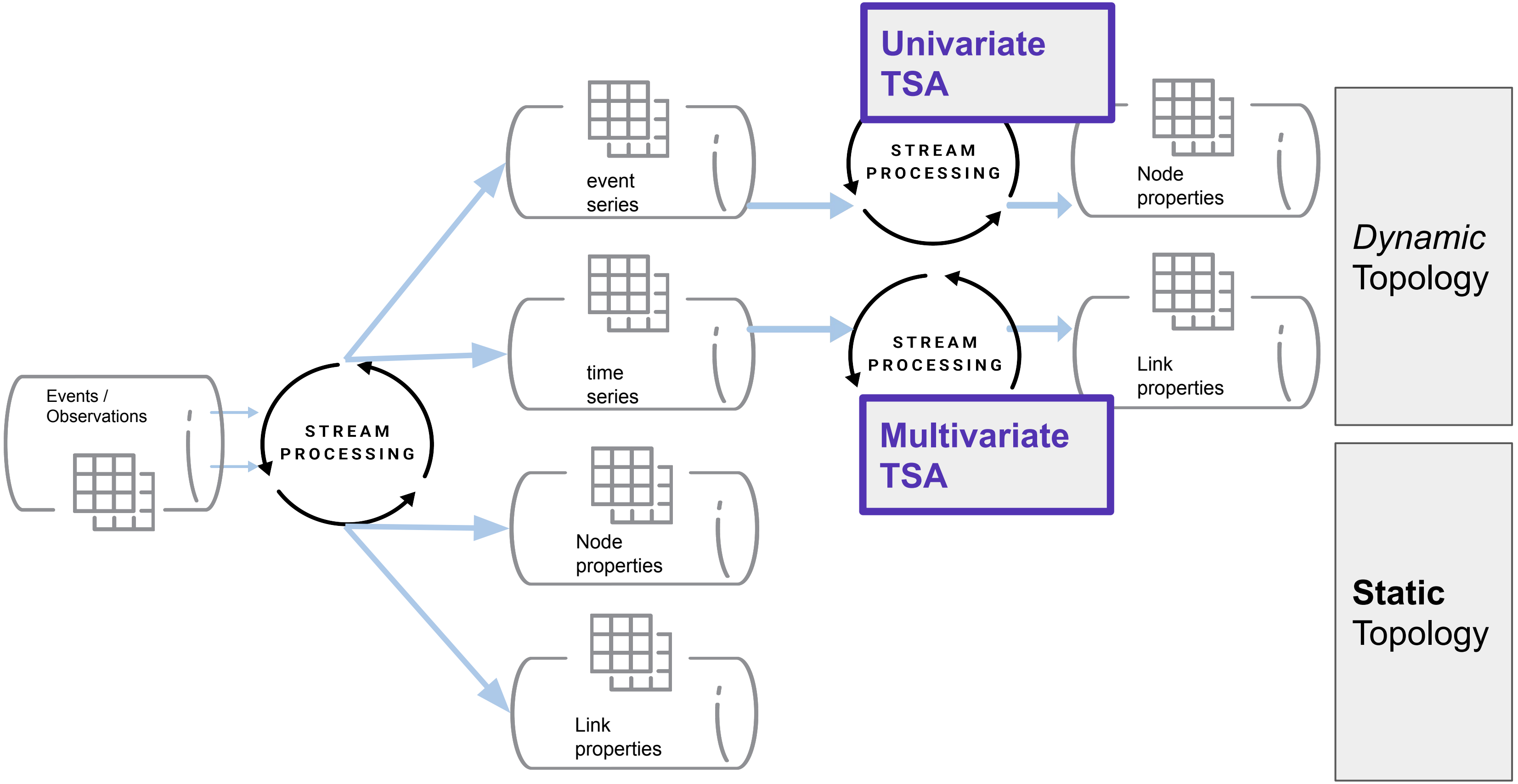


Table 1: Operators with their input and output types

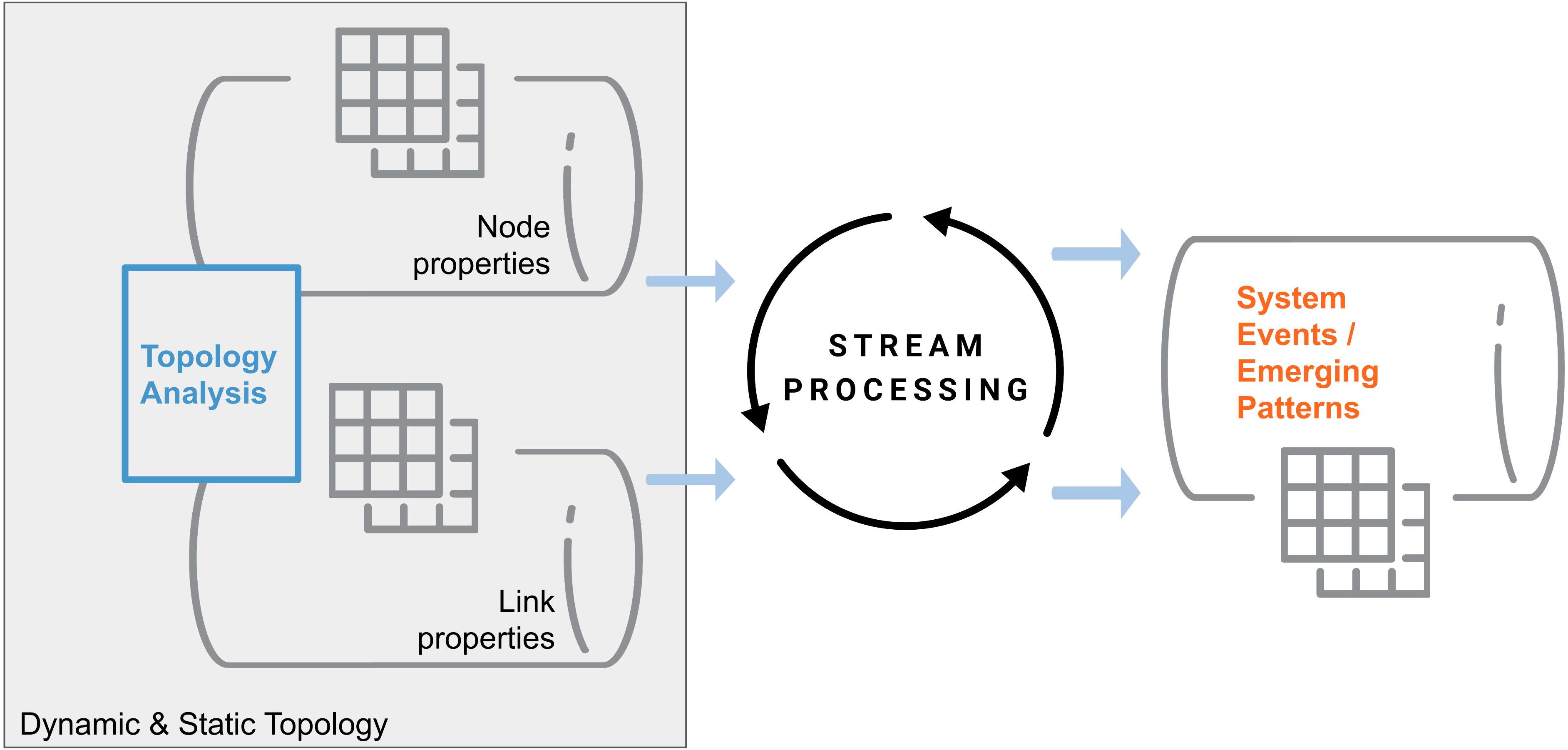
Operator	1st Input	2nd Input	Output
`filter`, `mapValue`	KStream		KStream
	KTable		KTable
`map`, `flatMap`	KStream		KStream
`groupBy` → `agg`	KStream		KTable
	KTable		KTable
`groupBy` + `windowBy` → `agg`	KStream		KTable
`inner-/left-/outer-join`	KStream	KStream	KStream
`inner-/left-/outer-join`	KTable	KTable	KTable

BIRTE '18, August 27, 2018, Rio de Janeiro, Brazil M. J. Sax, G. Wang, M. Weidlich, J.-C. Freytag

Multi Layer Stream Processing:
TSA to Reveal Hidden System Structures



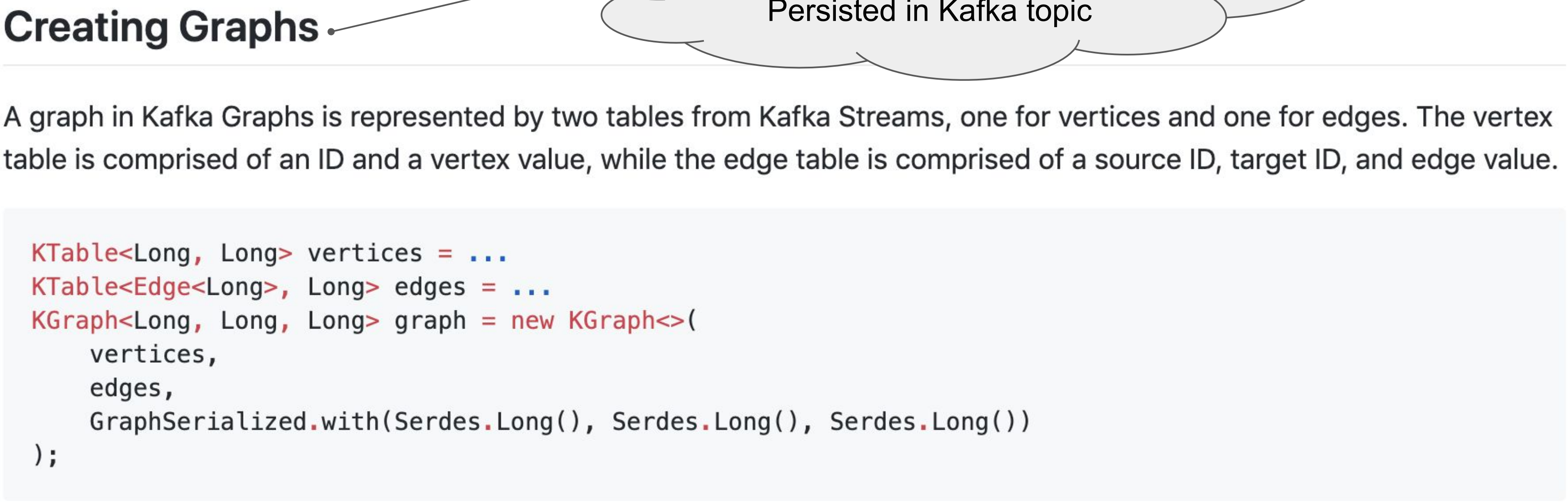
Complex Event Processing: For Complex Systems



Use the Table-Network Analogy: Kafka Graphs

large durable graphs:

Persisted in Kafka topic



Creating Graphs

A graph in Kafka Graphs is represented by two tables from Kafka Streams, one for vertices and one for edges. The vertex table is comprised of an ID and a vertex value, while the edge table is comprised of a source ID, target ID, and edge value.

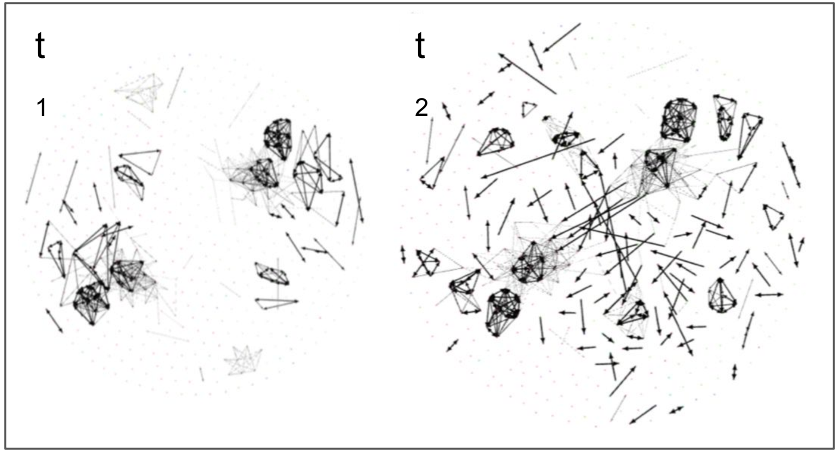
```
KTable<Long, Long> vertices = ...
KTable<Edge<Long>, Long> edges = ...
KGraph<Long, Long, Long> graph = new KGraph<>(
    vertices,
    edges,
    GraphSerialized.with(Serdes.Long(), Serdes.Long(), Serdes.Long())
);
```

https://github.com/rayokota/kafka-graphs

Sliding Windows: Define the Temporal Graphs

In some use cases, we don’t want to keep the node and link data in topics:

- nodes aren’t always linked
- changes are very fast
- focus on activation patterns, rather than on underlying structure



It is fine to calculate the correlation links and the topology metrics on the fly, just for a given time window.

JGraphT

commons Math™

Back to Streams …

Architecture:
Identify Patterns &
Buildingblocks

Let's look into 3 examples:

(1) Linear flow ...

(2) Bi-directional flow ...

(3) Complex process flows ...

Let's look into 3 examples:

(1) Linear flow ...

(2) Bi-directional flow ...

(3) Complex process flows ...

Easy:

1. reusable
2. scalable
3. ready to use
4. ready to improve

⇒ Target for simplification ...

A Standardized Processing Procedure for Episodes

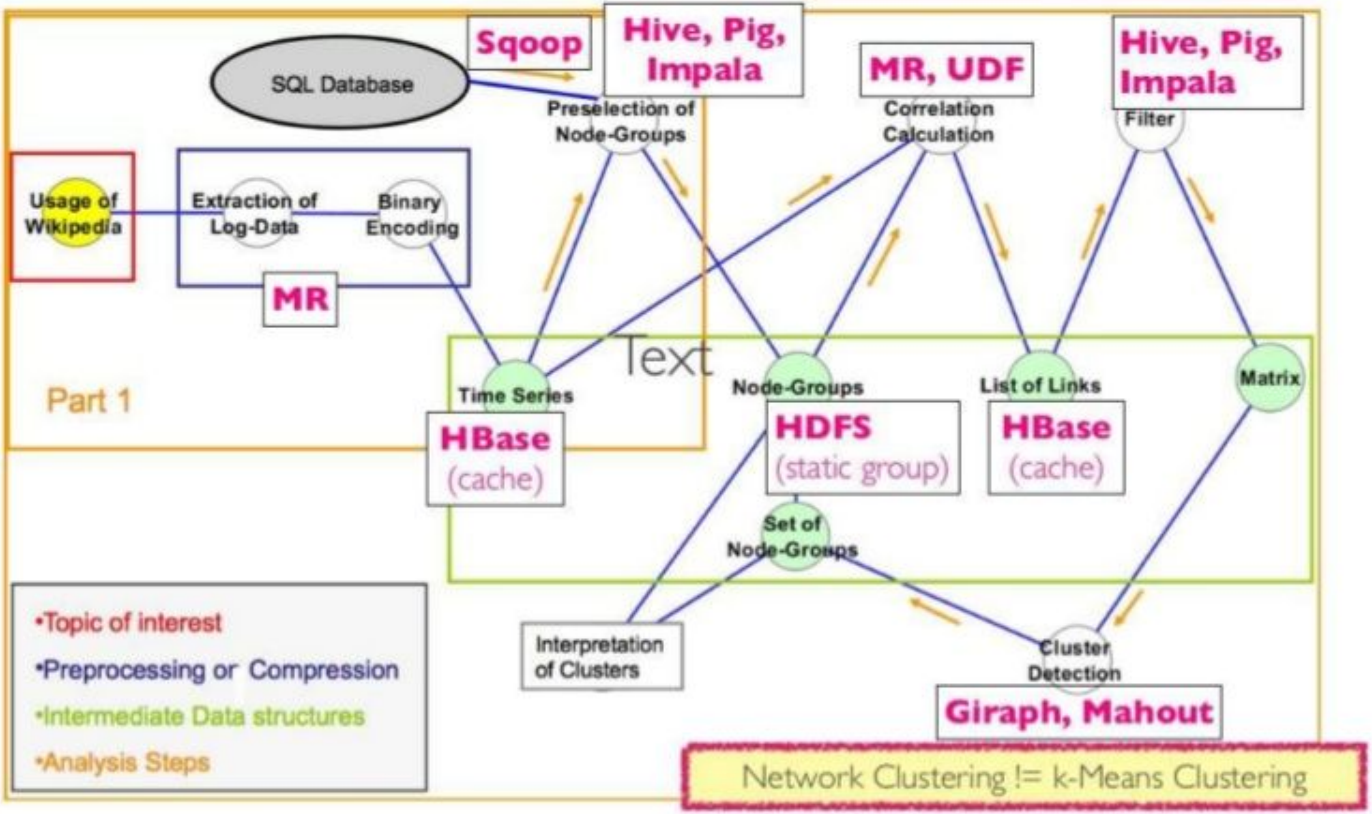
used for social media analysis on Hadoop:

- The predecessor of OpenTSx is Hadoop.TS (https://www.researchgate.net/publication/269687614_Hadoop_TS_Large-Scale_Time-Series_Processing)

- Hadoop.TS used a variety of Hadoop ecosystem projects (Sqoop, Flume, Hive, Spark, Yarn, HDFS, Solr, HBase, Impala, Mahout, Giraph)

- Managing the data flow at scale was **possible**, but complex.

OVERVIEW - DATA FLOW



Network Clustering != k-Means Clustering

This example illustrate the variety
of interconnected components from our implementation
in the Hadoop ecosystem.

The resulting complexity of a solution can become a blocker!

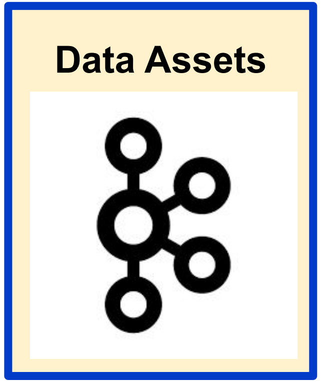
Architects have to simplify …

… and the Kafka Ecosystem helps you on this journey!

KEEP
CALM
AND
Simplify
KeepCalmAndPosters.com

Building Blocks:

Data flows are no
longer transient.
The event log acts as
single source of truth.



Data Assets

*Paradigm Shift in
Data Management*

Building Blocks:

Domain specific logic is implemented in reusable components:

UDFs:

commons Math

JGraphT

Domain Driven Design

Data flows are no longer transient.
The event log acts as single source of truth.

Data Assets

Paradigm Shift in Data Management

Building Blocks:

Domain specific logic is implemented in reusable components:

UDFs:

commons Math

JGraphT

Domain Driven Design

Source Connectors Integrate Input side …

Kafka Connect

APACHE PLC4X

Legacy and Future Systems

Data flows are no longer transient.
The event log acts as single source of truth.

Data Assets

Paradigm Shift in Data Management

Building Blocks:

Domain specific logic is implemented in reusable components:

UDFs:

commons Math

JGraphT

Domain Driven Design

Source Connectors
Integrate Input side …

Kafka Connect

APACHE PLC4X

Legacy and Future Systems

Data flows are no longer transient.
The event log acts as single source of truth.

Data Assets

Paradigm Shift in Data Management

Sink Connectors
Integrate Output side …

Kafka Connect

influxdb

neo4j

Special Purpose Systems

Demo: OpenTSx

https://github.com/kamir/OpenTSx

Generate some observations.

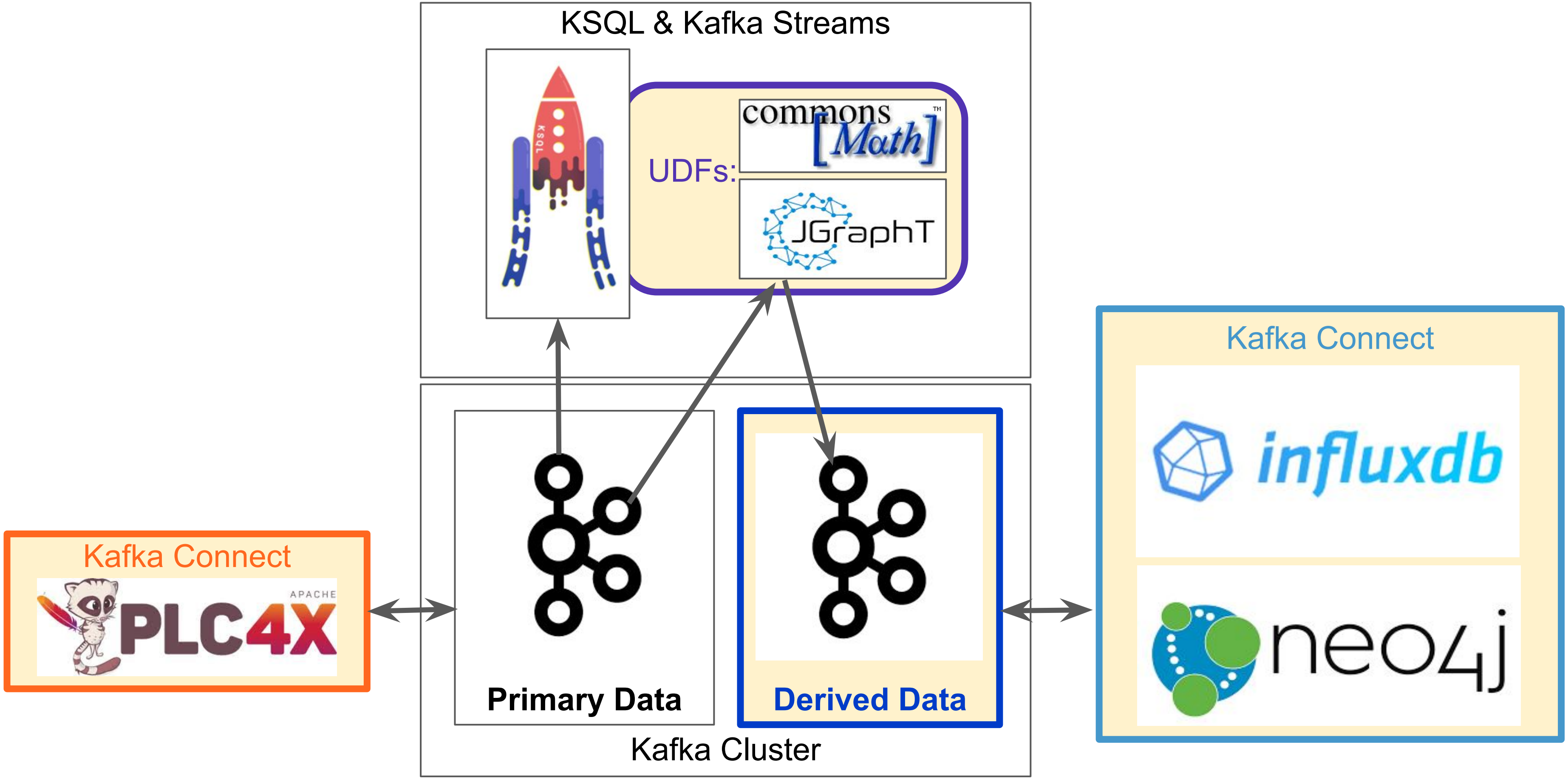
Form an episode (using windowing functions).

Apply some time series processing procedures on the stream of episodes.
>>> Automatically define a KStreams application via KSQL statement using UDFs.

Complex procedures are composed from a set of fundamental building blocks.

>>> Deploy ksqlDB query to your streaming data in a Kafka cluster.

Kafka: A Platform for Complex Event Processing



Summary:

Because Kafka is a scalable & extensible platform it fits well for complex event processing in any industry on premise and in the cloud.

Kafka ecosystem provides extension points for any kind of domain specific or custom functionality - from advanced analytics to real time data enrichment.

Complex solutions are composed from a few fundamental building blocks:

What to do next?

(A) Identify **relevant main flows** and **processing patterns** in your project.

(B) Identify or implement source / sink **connectors** and establish 1st flow.

(C) Implement custom transformations as **Kafka independent components**.

(D) Integrate the processing topology as Kafka Streams application:

- (a) Do you apply standard transformations and joins (for enrichment)?
- (b) Is a special treatment required (advanced analysis)?
- (c) Do you need special hardware / external services (AI/ML for classification)?

(E) Share your connectors and UDFs with the growing Kafka community.

(F) Iterate, add more flows and more topologies to your environment.

THANK YOU

mirko@confluent.io