Berlin, Sep 11-13, 2017

I^2

INTERACTIVE REAL-TIME VISUALIZATION FOR STREAMING DATA WITH APACHE FLINK AND APACHE ZEPPELIN

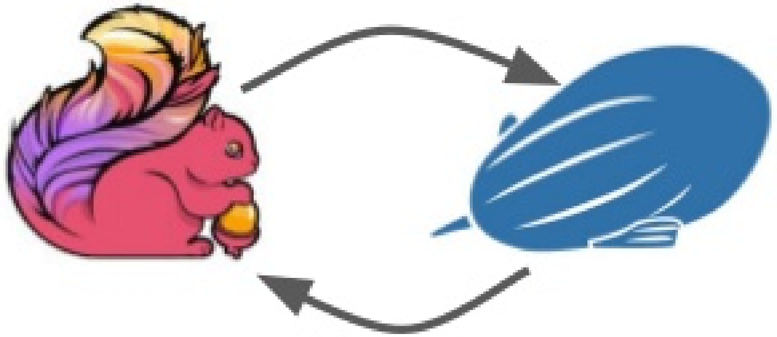
JONAS TRAUB
RESEARCH ASSOCIATE,
TECHNISCHE UNIVERSITÄT BERLIN

PHILIPP GRULICH
RESEARCH ASSISTANTS,
GERMAN RESEARCH CENTRE FOR ARTIFICIAL INTELLIGENCE

It’s worth it to postpone checking your mails!

We connect Flink and Zeppelin to visualize data-streams in real-time.

We make front-end settings available to running Flink jobs.



We adapt running Flink jobs to visualization requirements.

We reduce the amount of processed and transferred data while providing loss-free plots.

i.e., we visualize 12.000 events per second without crashing the front end.

We enable visualization-driven development of Flink jobs.

two types of interactivity

(i) through code changes

change and deploy the code of analysis pipelines and corresponding result visualizations in a one-click fashion

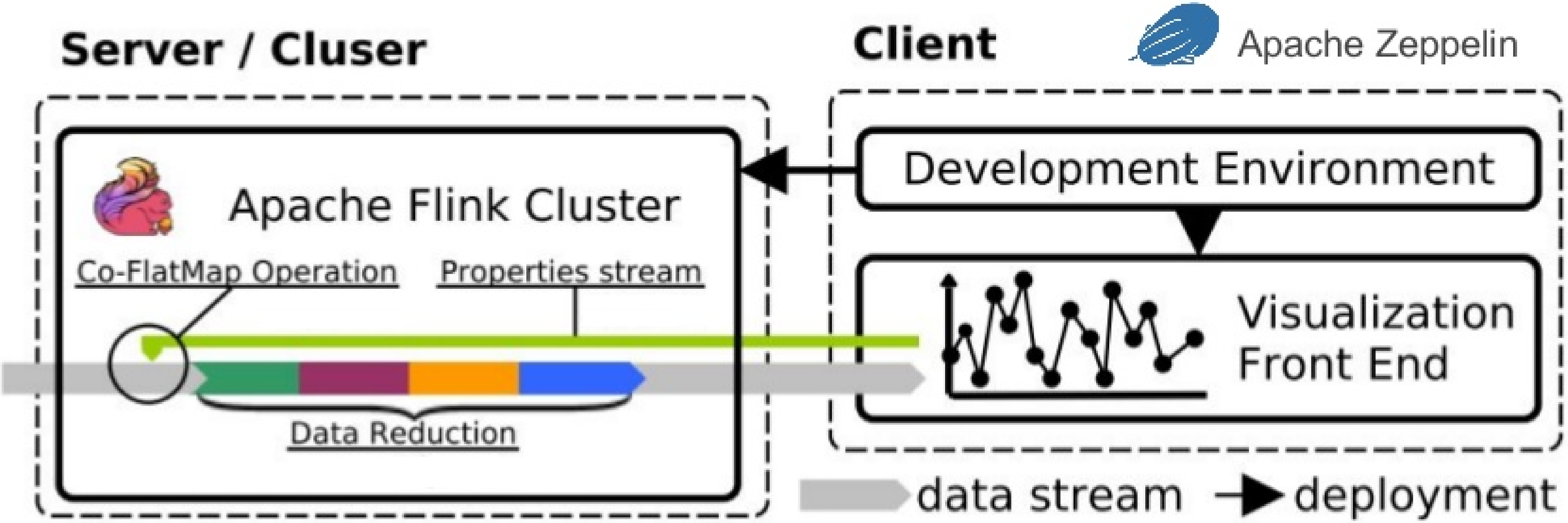
Rapid data-driven development of data analysis pipelines

(ii) through an interactive visualization GUI

change visualization properties (e.g. the zoom level in a map) while the underlying Flink job adapts at runtime

Reduces processed and transferred data while still providing loss-free visualizations

Architecture Overview



I² - Live Demonstration

The DEBS 2013 Grand Challenge

Christopher Mutschler, Holger Ziekow, Zbigniew Jerzak; DEBS'13

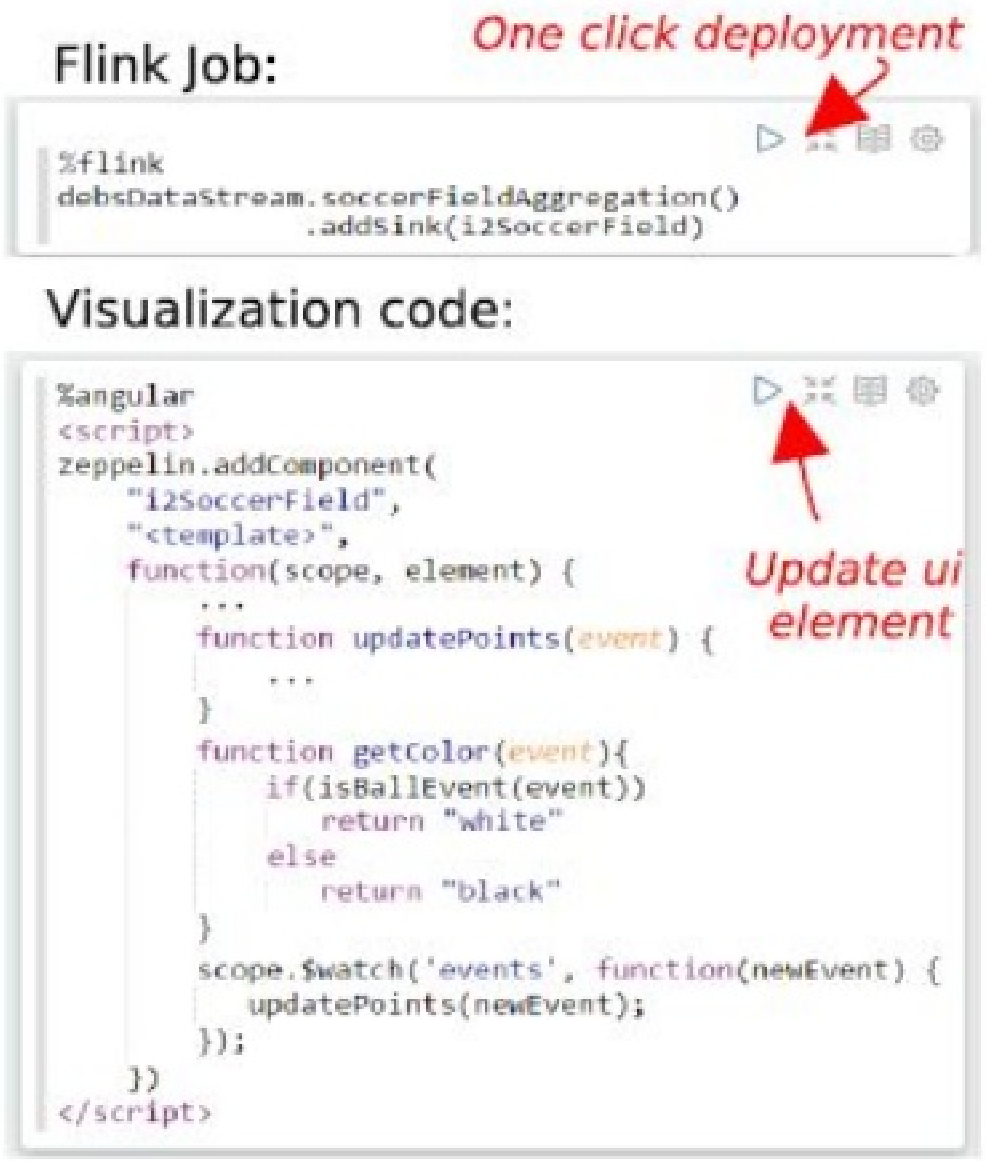
Data:

- Sensor data from a football match
  (speed, acceleration, and position of the ball and players)
- Up to 2000 Hz frequency
- roughly 12.000 data points per second

Demo

I^2

Development Environment



Flink Job:

One click deployment

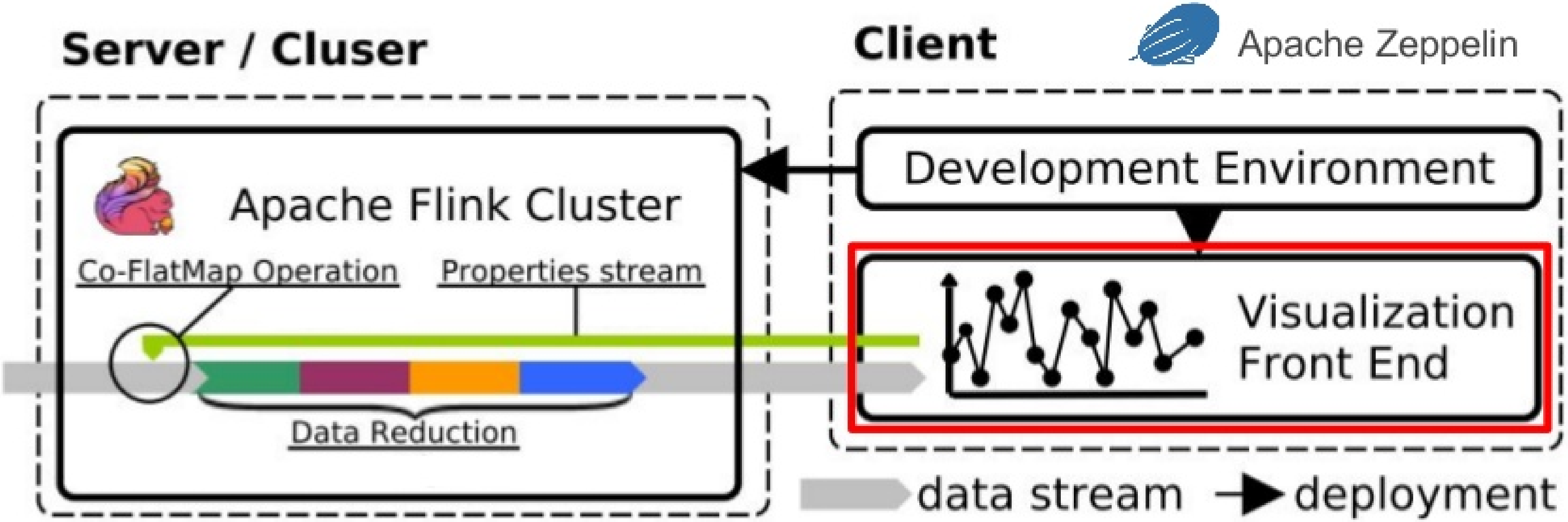
```
%flink
debsDataStream.soccerFieldAggregation()
                 .addSink(i2SoccerField)
```

Visualization code:

```
%angular
<script>
zeppelin.addComponent(
     "i2SoccerField",
     "<template>",
     function(scope, element) {
          ...
          function updatePoints(event) {
               ...
          }
          function getColor(event){
               if(isBallEvent(event))
                   return "white"
               else
                   return "black"
          }
          scope.$watch('events', function(newEvent) {
             updatePoints(newEvent);
          });
     })
</script>
```

Update ui element

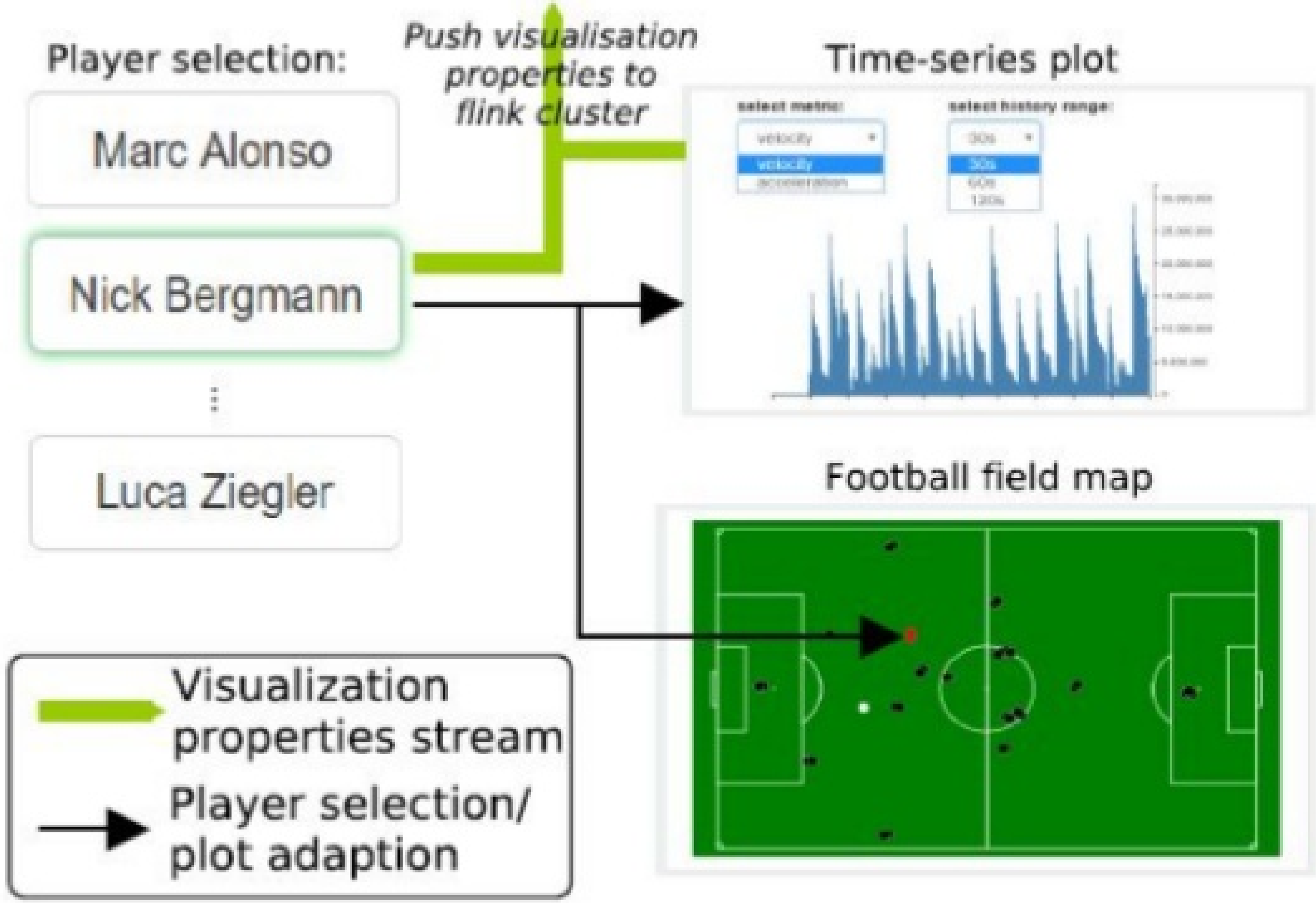
Architecture Overview



Visualization Front End

UI components push visualization properties to the flink cluster

Stream only visible data points towards the front end



Player selection:

Marc Alonso

Nick Bergmann

⋮

Luca Ziegler

Push visualisation properties to flink cluster

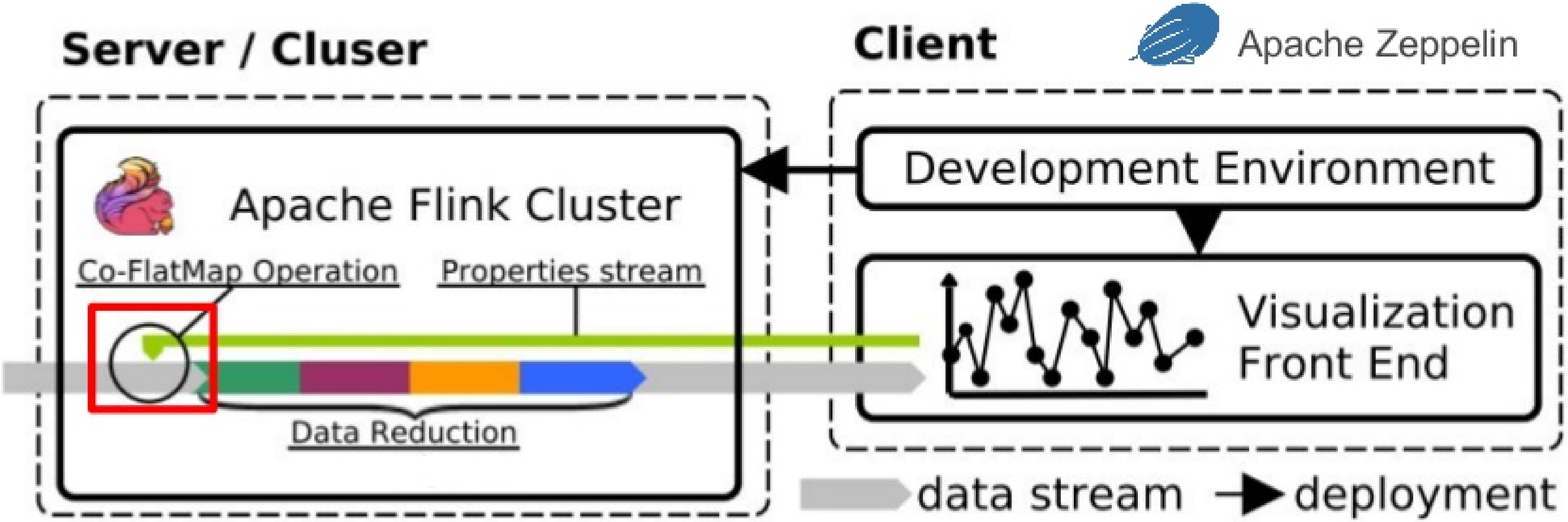
Time-series plot

Football field map

— Visualization properties stream

→ Player selection/ plot adaption

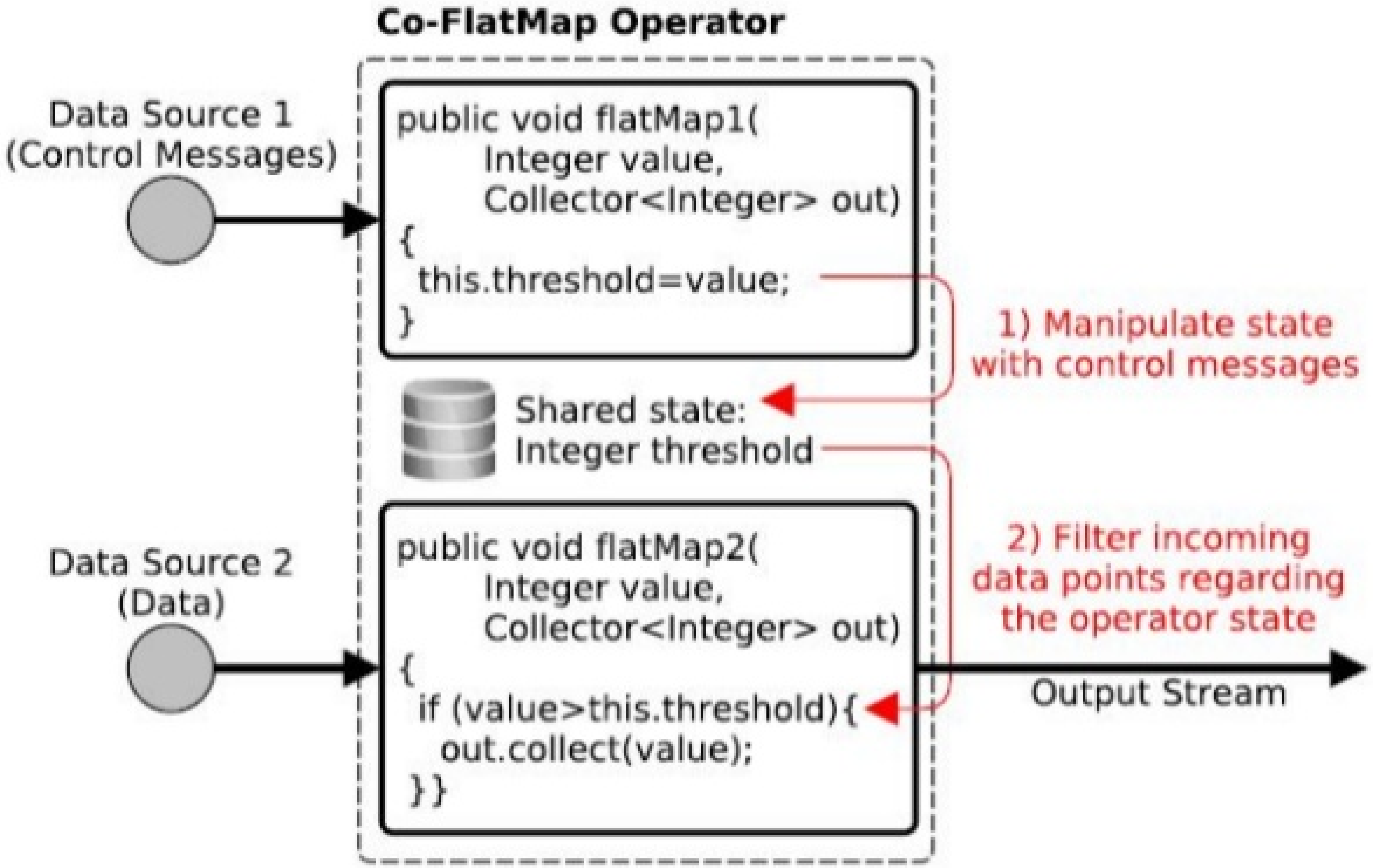
Architecture Overview



Adaptive Operator Example: Adapt to Visualization Settings

Control Messages:
Updates to a filter threshold

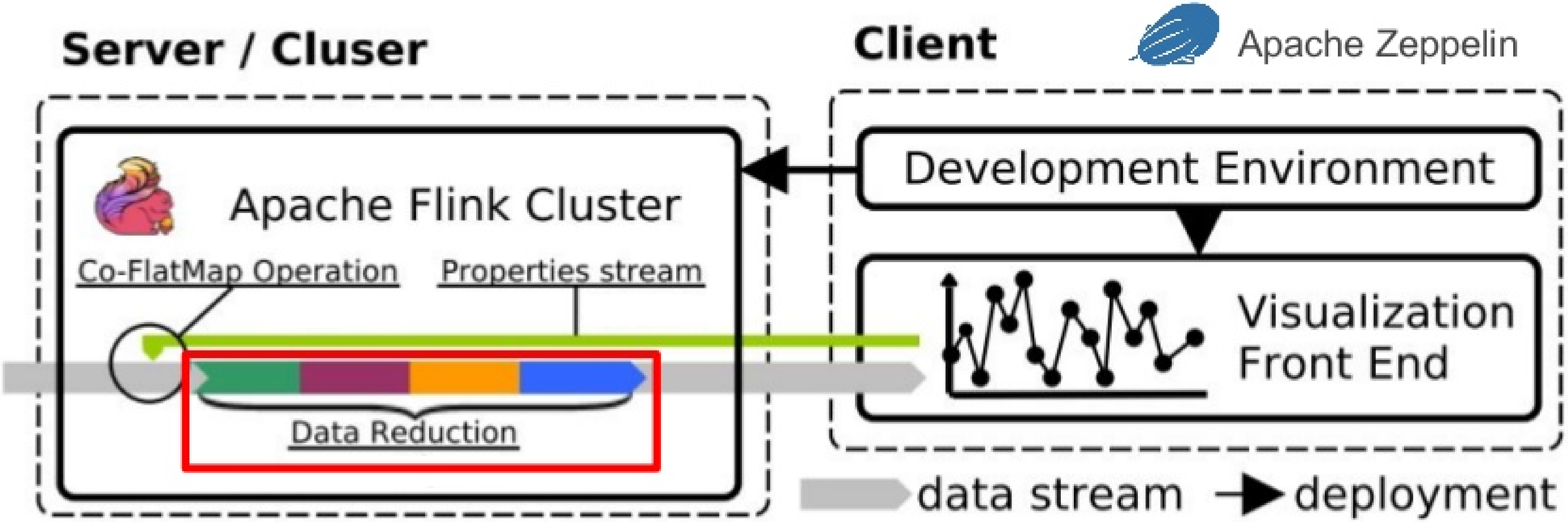
Data Processing:
Filter according to current threshold



```
public void flatMap1(
        Integer value,
        Collector<Integer> out)
{
 this.threshold=value;
}
```

```
public void flatMap2(
        Integer value,
        Collector<Integer> out)
{
 if (value>this.threshold){
   out.collect(value);
 }}
```

Architecture Overview



Loss-free Visualization with Reduced Costs

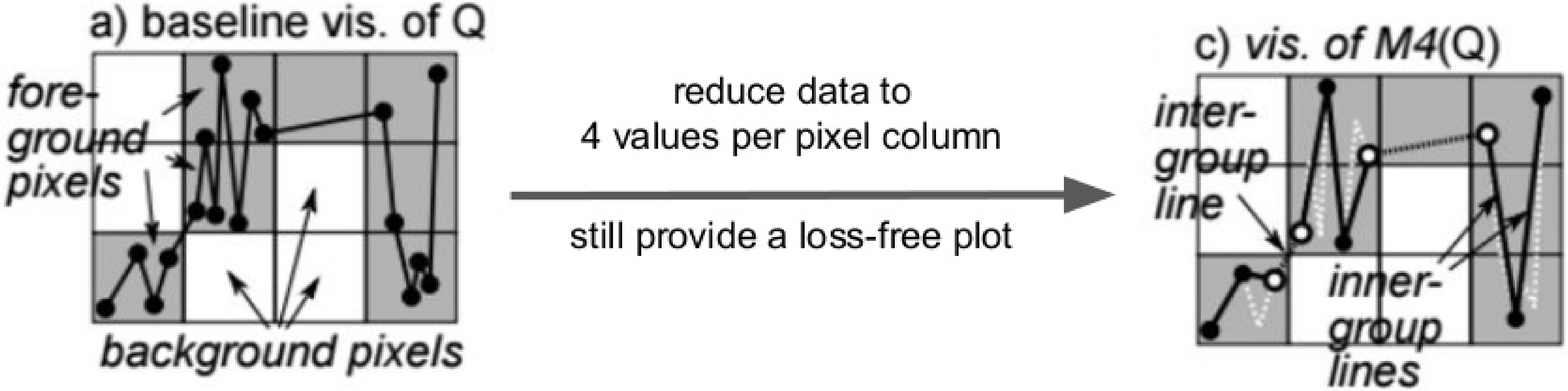
M4: A Visualization-Oriented Time Series Data Aggregation

Uwe Jugel, Zbigniew Jerzak,
Gregor Hackenbroich
SAP AG
Chemnitzer Str. 48, 01187 Dresden, Germany

Volker Markl
Technische Universität Berlin
Straße des 17. Juni 135
10623 Berlin, Germany

VLDB
2014

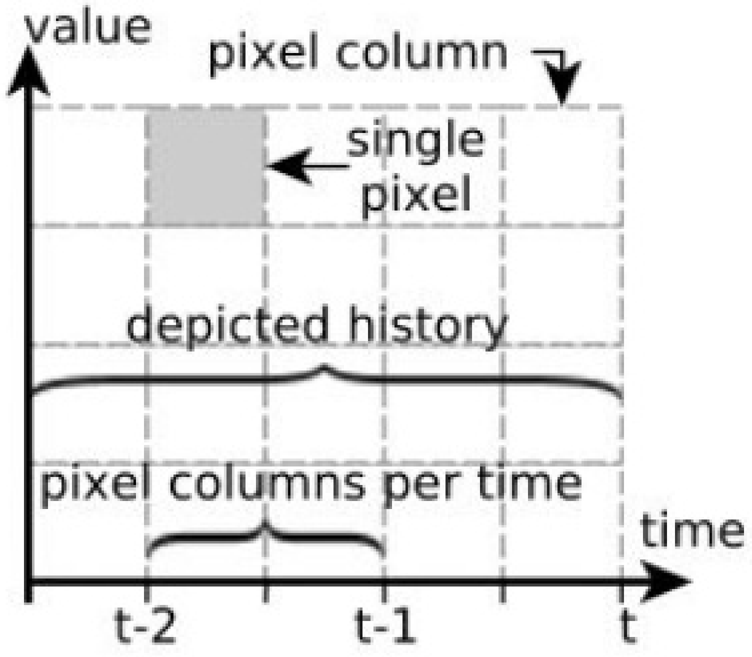
Big data time series often contain more data per pixel than we can display.



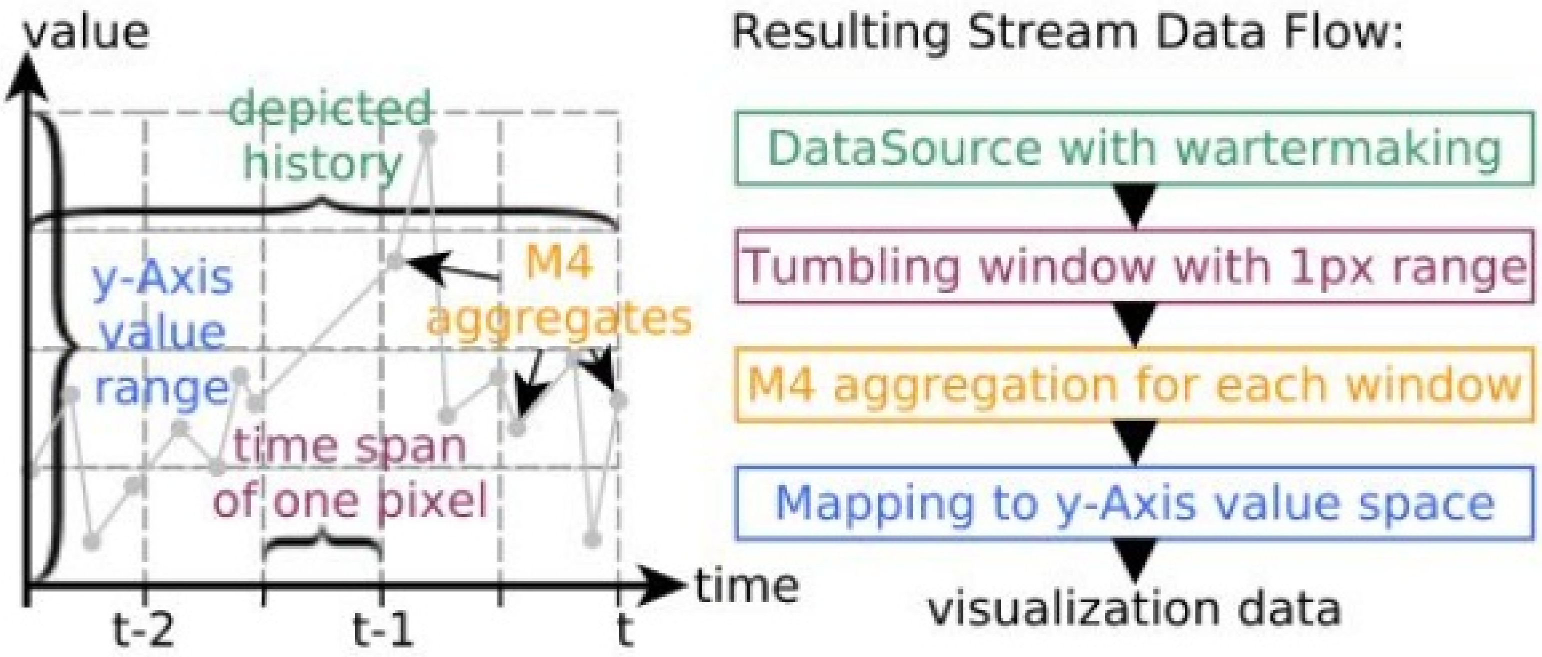
reduce data to
4 values per pixel column

still provide a loss-free plot

I² introduces a streaming ready version of M4



(a) The tradeoff between depicted history and plot precision.



Resulting Stream Data Flow:

DataSource with wartermaking

Tumbling window with 1px range

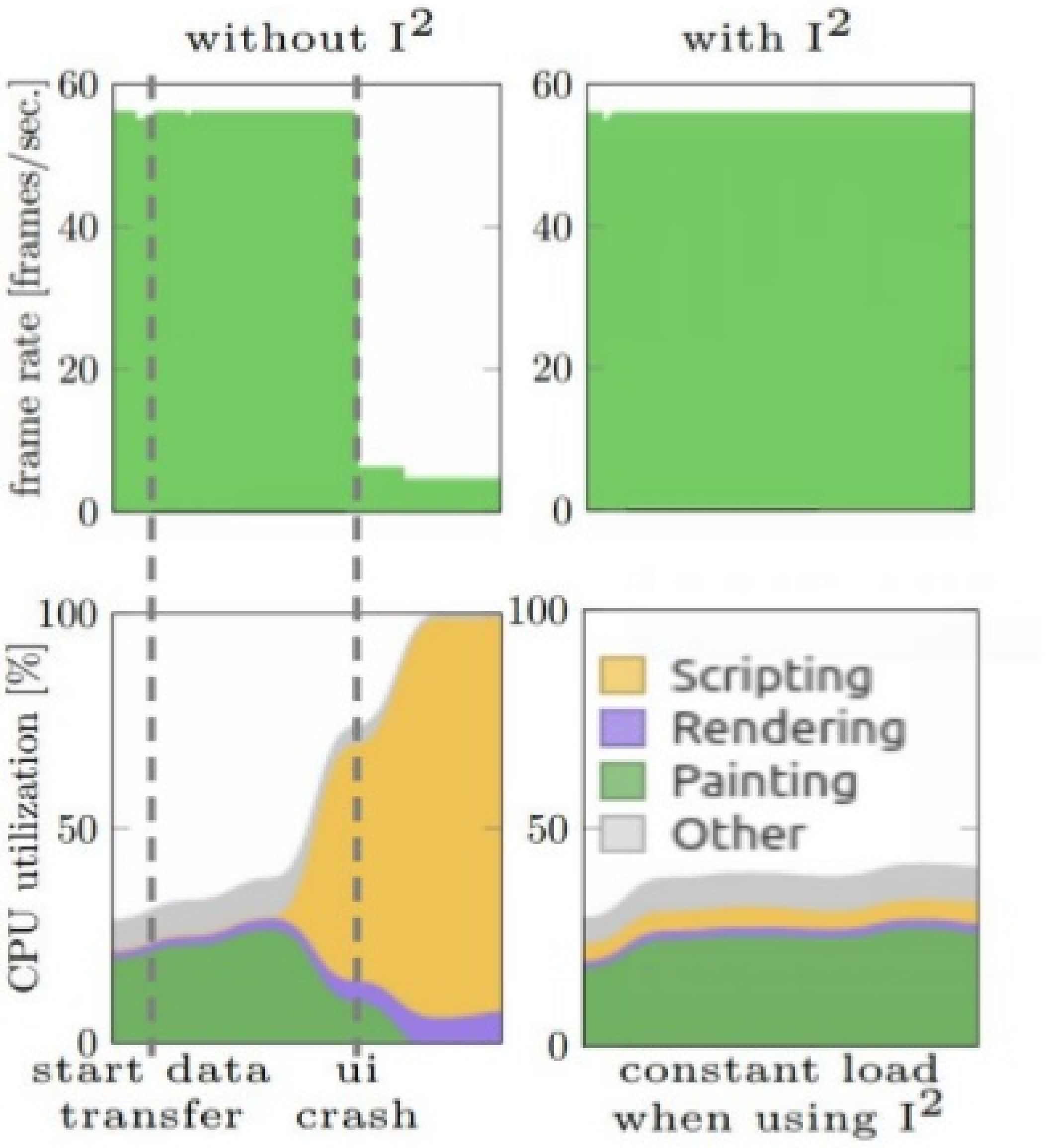
M4 aggregation for each window

Mapping to y-Axis value space

visualization data

(c) Deriving a stream data flow program for the real-time visualization of time-series data with M4.

The Impact of I² on the Visualization Front End



Thank you for joining our Session!

Summary

- Live visualization with Flink and Zeppelin
- Adapt jobs to changed setting at runtime
- Reduce data transfer and processing effort w/o quality-loss
- Support the development with live-visualization

Open Source

I^2 github.com/TU-Berlin-DIMA/i2

hub.docker.com/r/tuberlindima/i2

This talk and I^2 are supported by the EU Horizon 2020 Projects Proteus (687691) and Streamline (688191) and by the German Ministry for Education and Research as Berlin Big Data Center (01IS14013A) and Software Campus (01IS12056)