VISP TESTBED

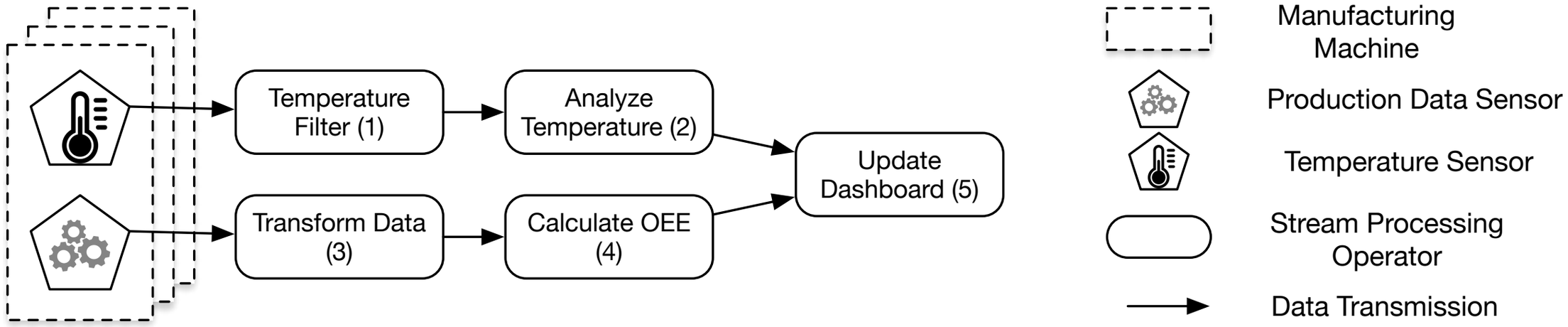
A Toolkit for Modeling and Evaluating Resource Provisioning Algorithms for Stream Processing Applications

---

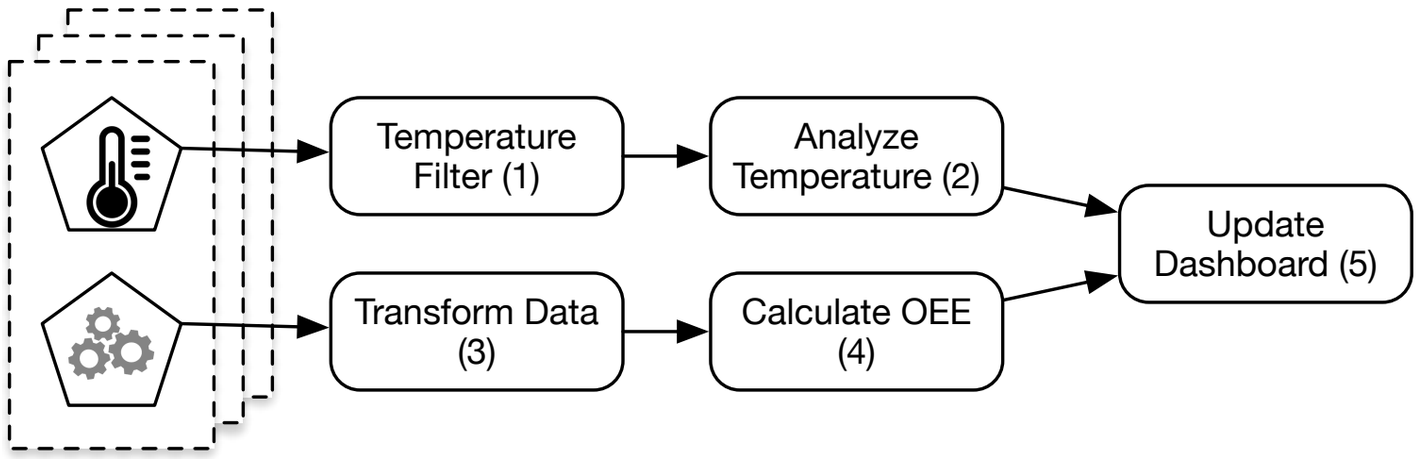
Christoph Hochreiner

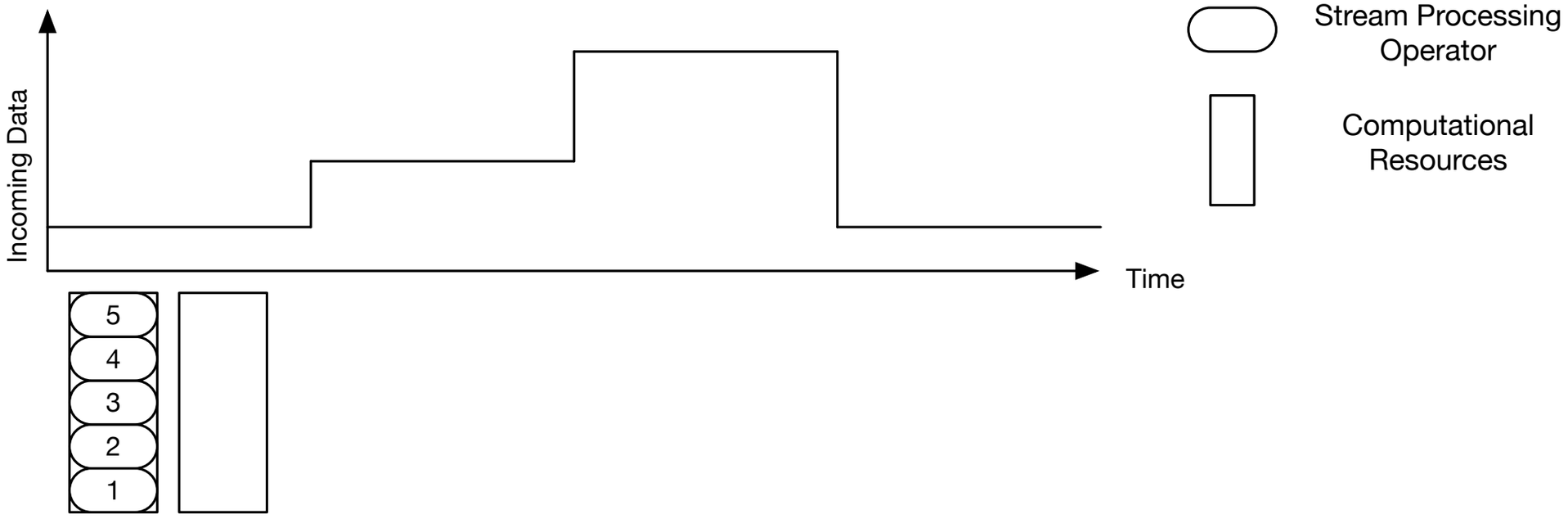
Motivation

Stream Processing Application

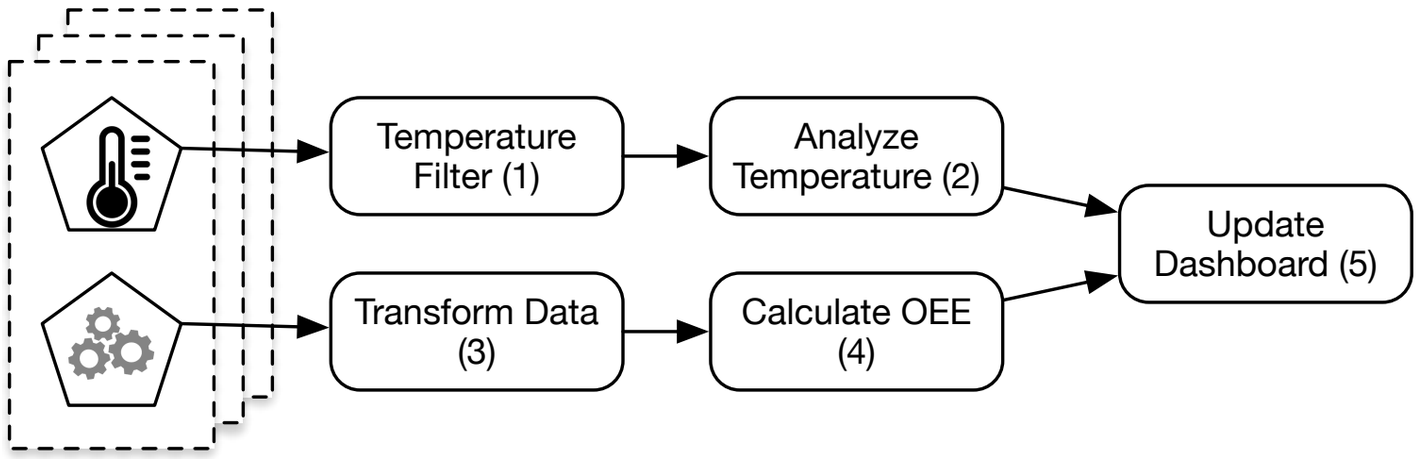


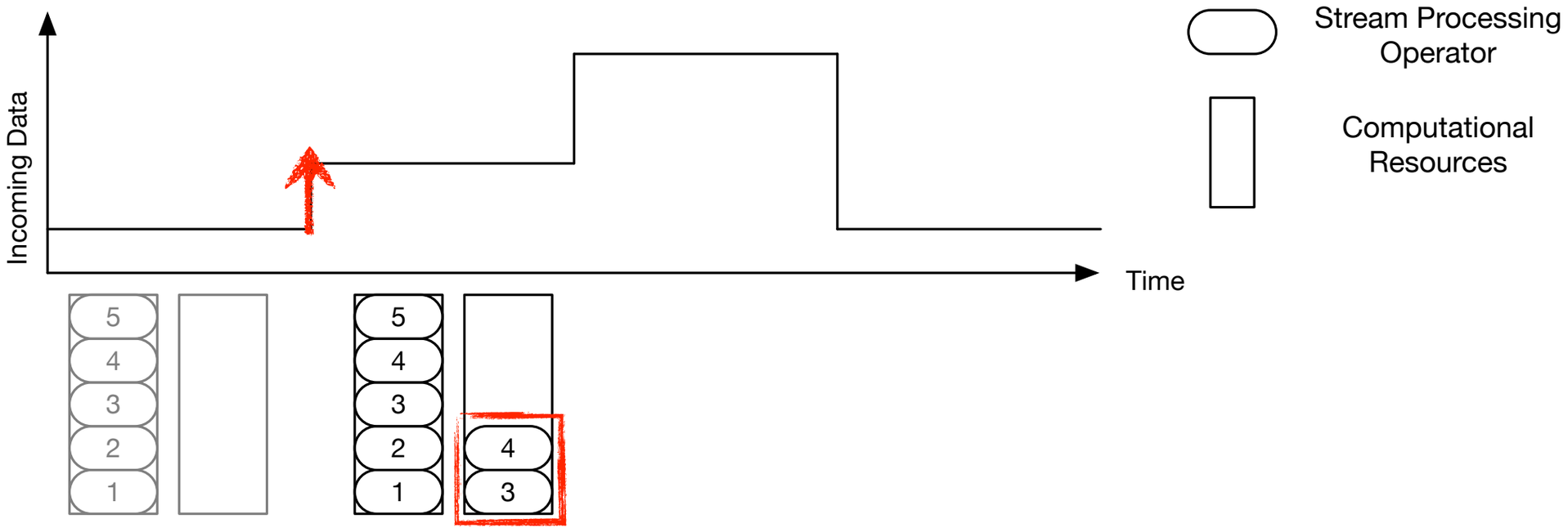
Stream Processing Application Deployment



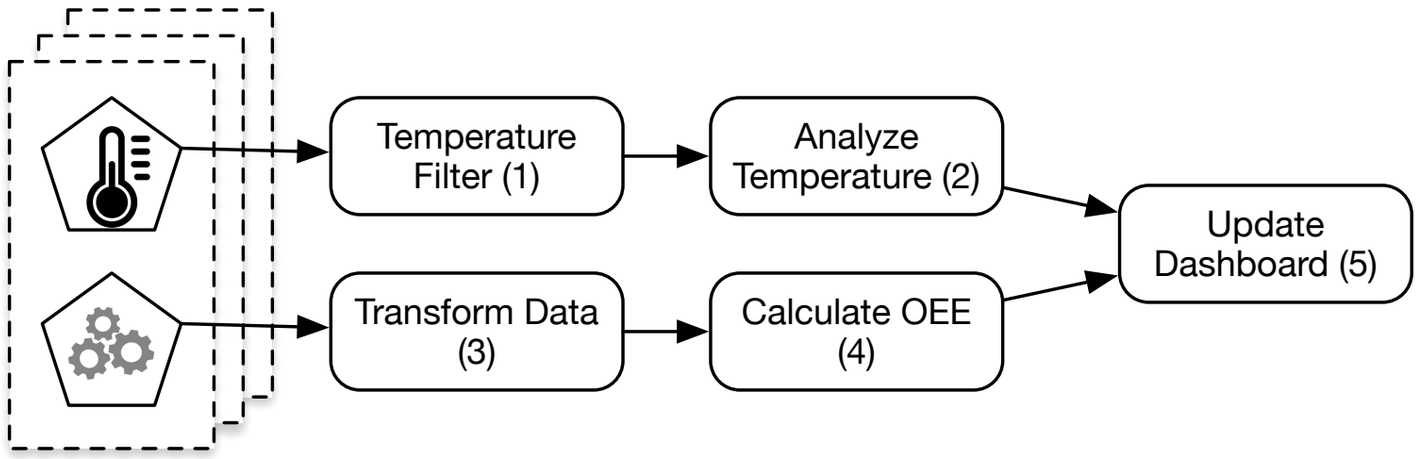


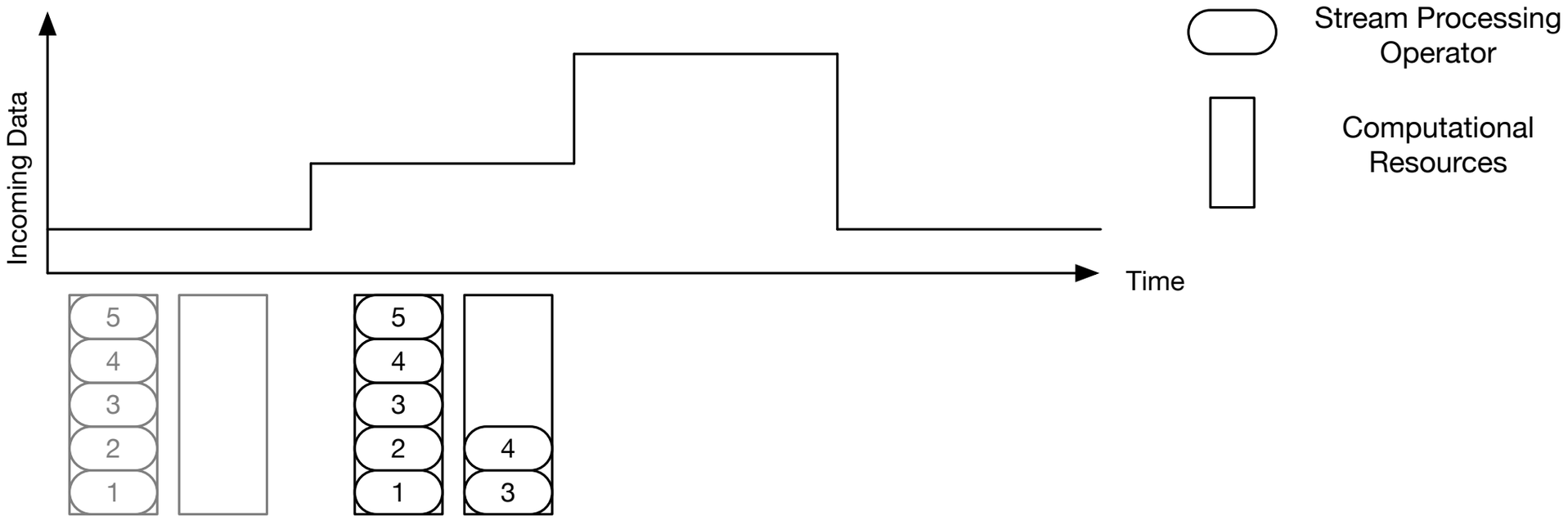
Stream Processing Application Deployment



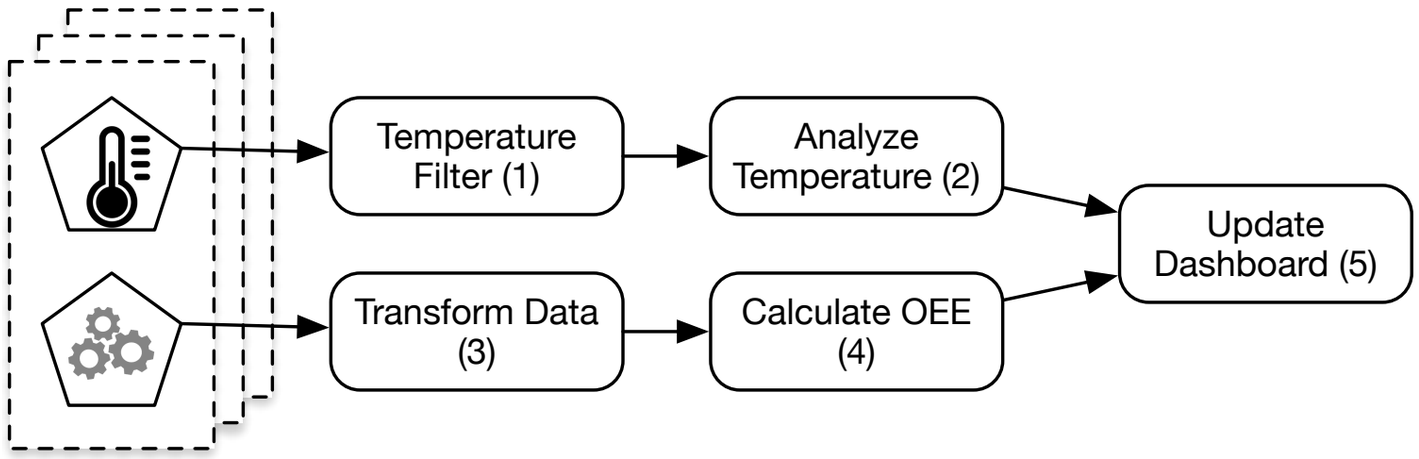


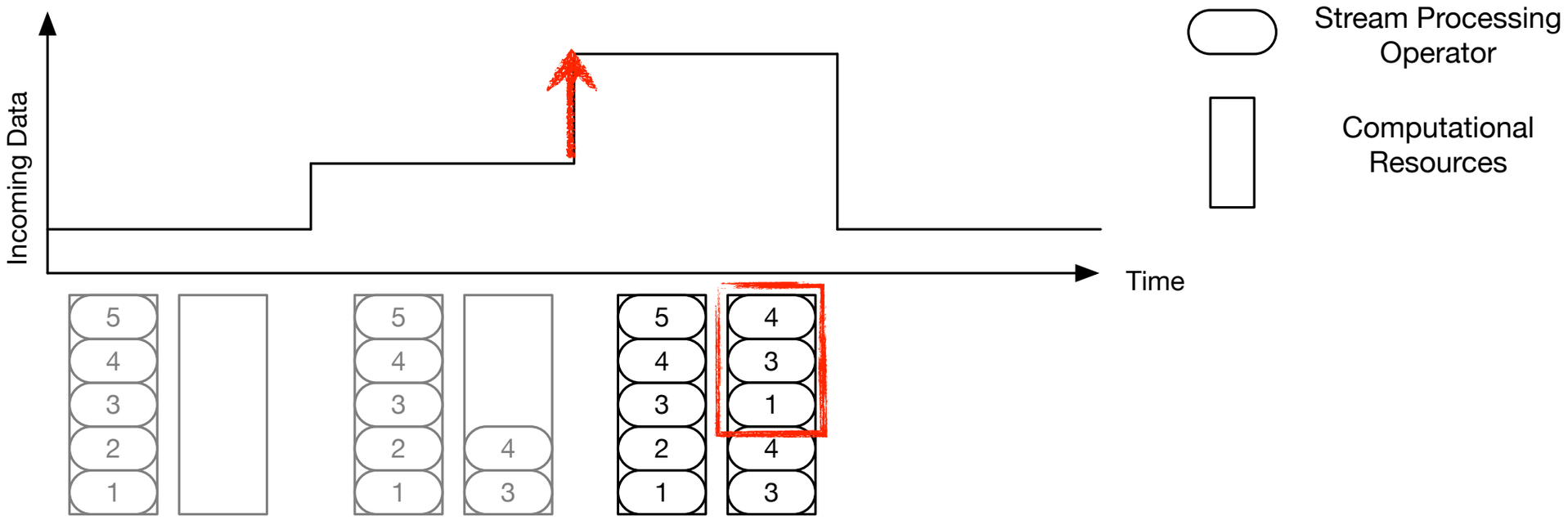
Stream Processing Application Deployment



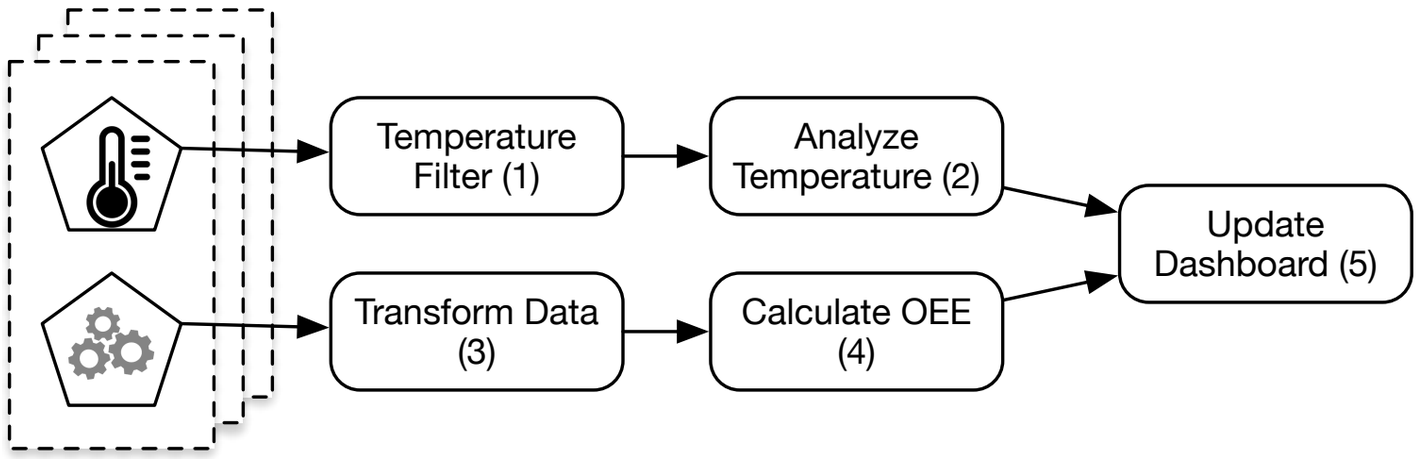


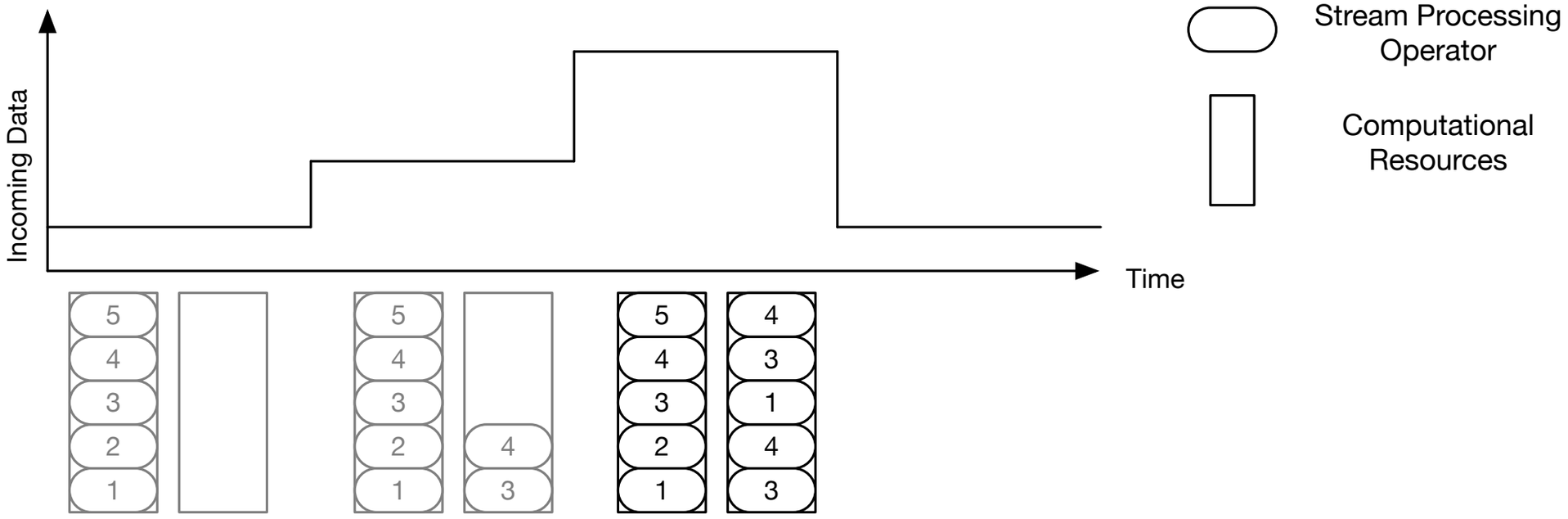
Stream Processing Application Deployment



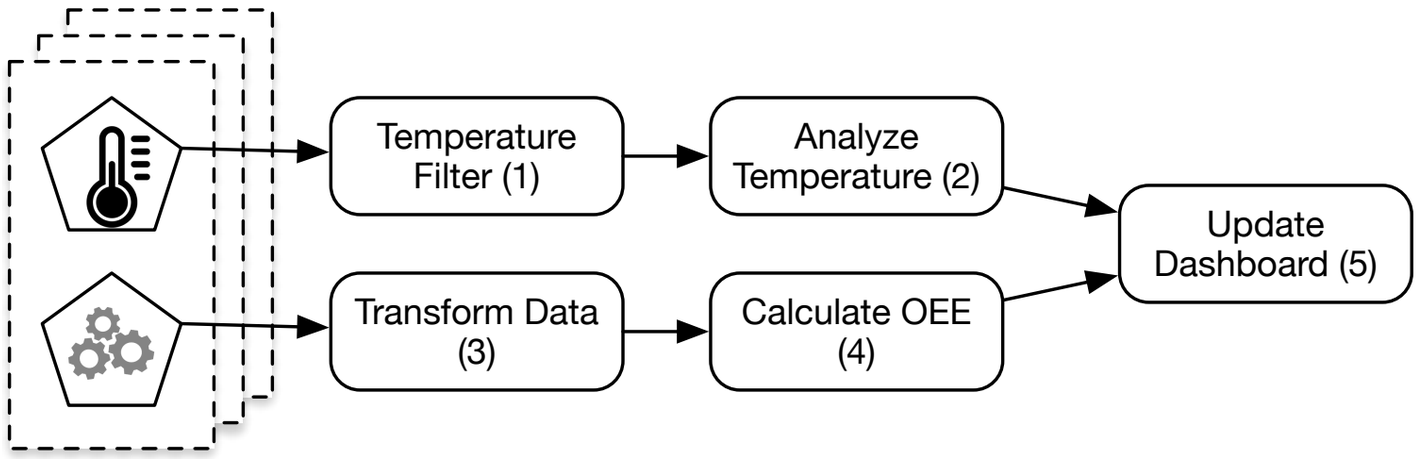


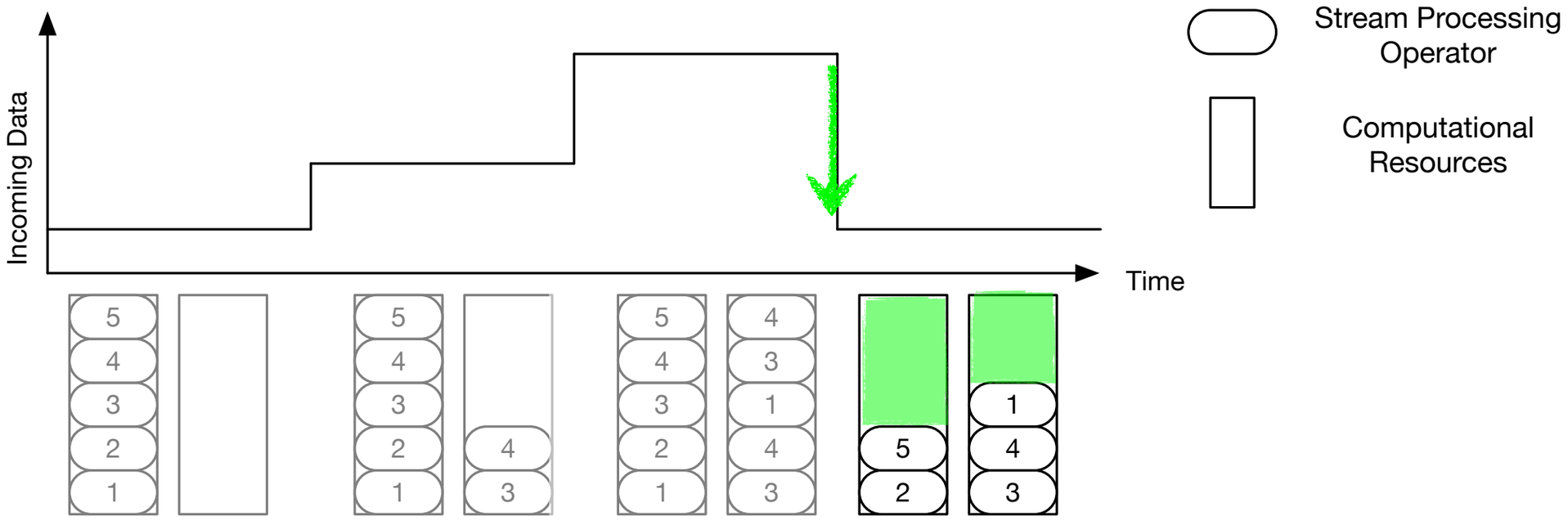
Stream Processing Application Deployment



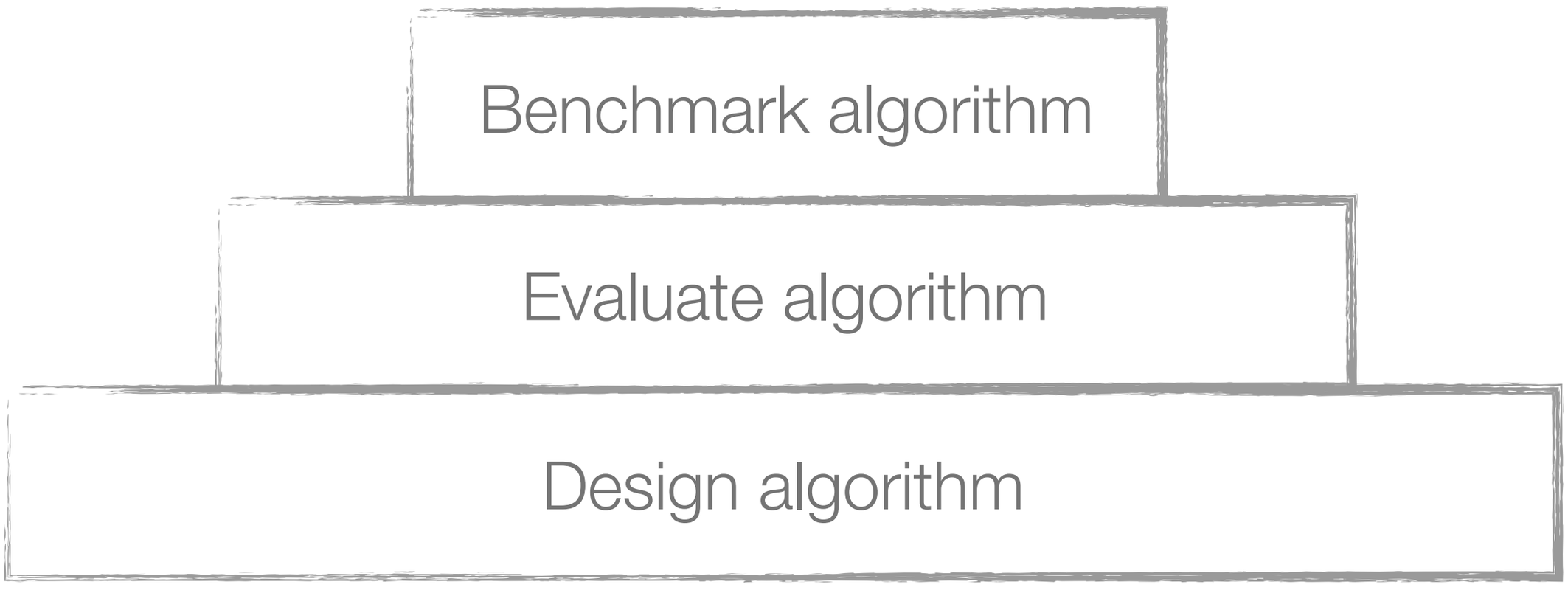


Stream Processing Application Deployment



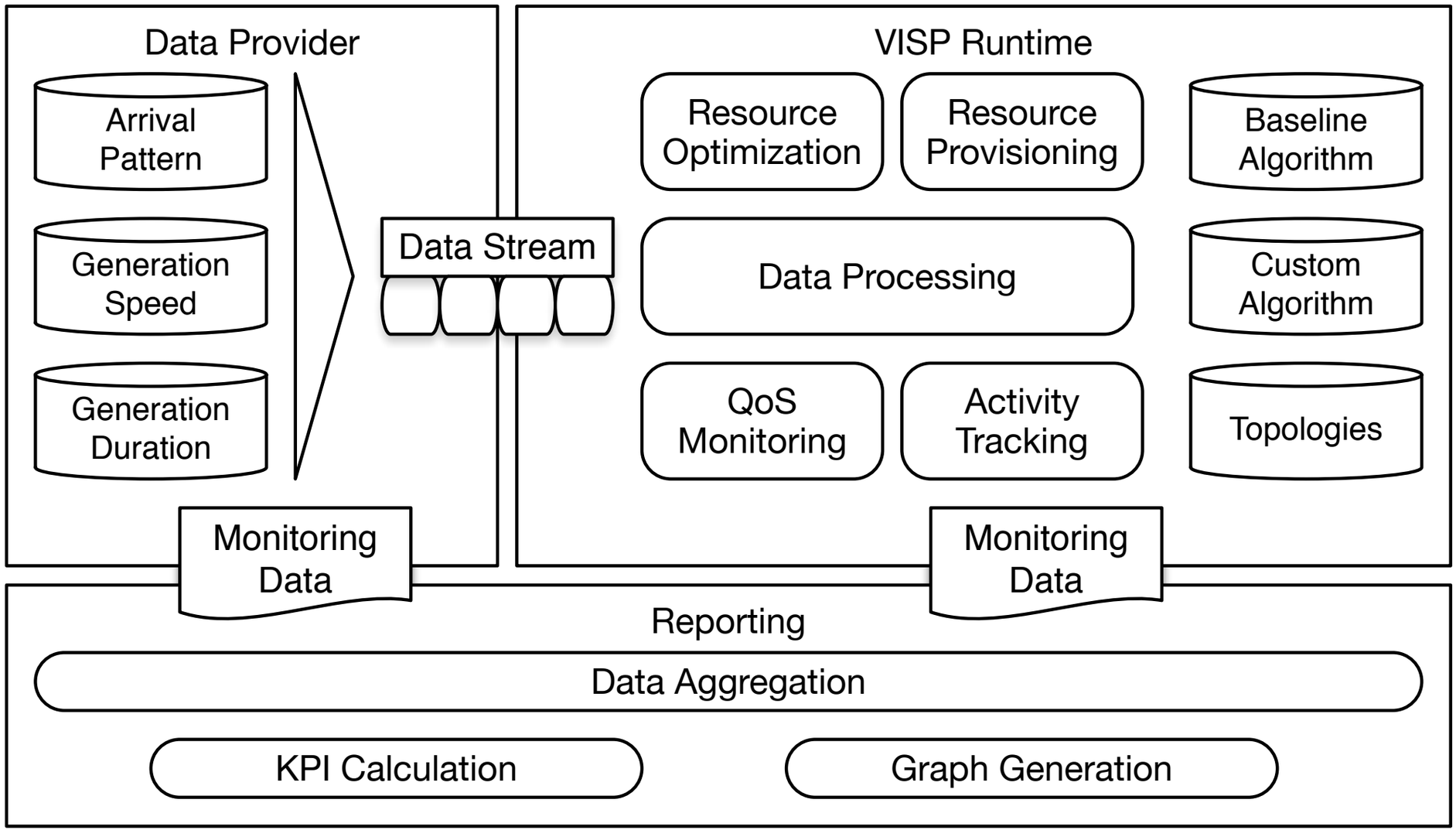


Steps to design resource provisioning algorithms

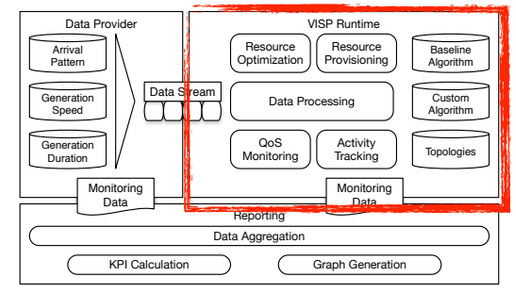


VISP Testbed

VISP Testbed

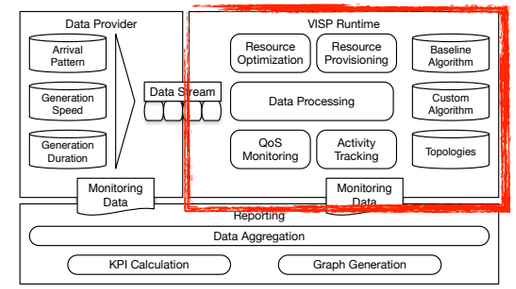


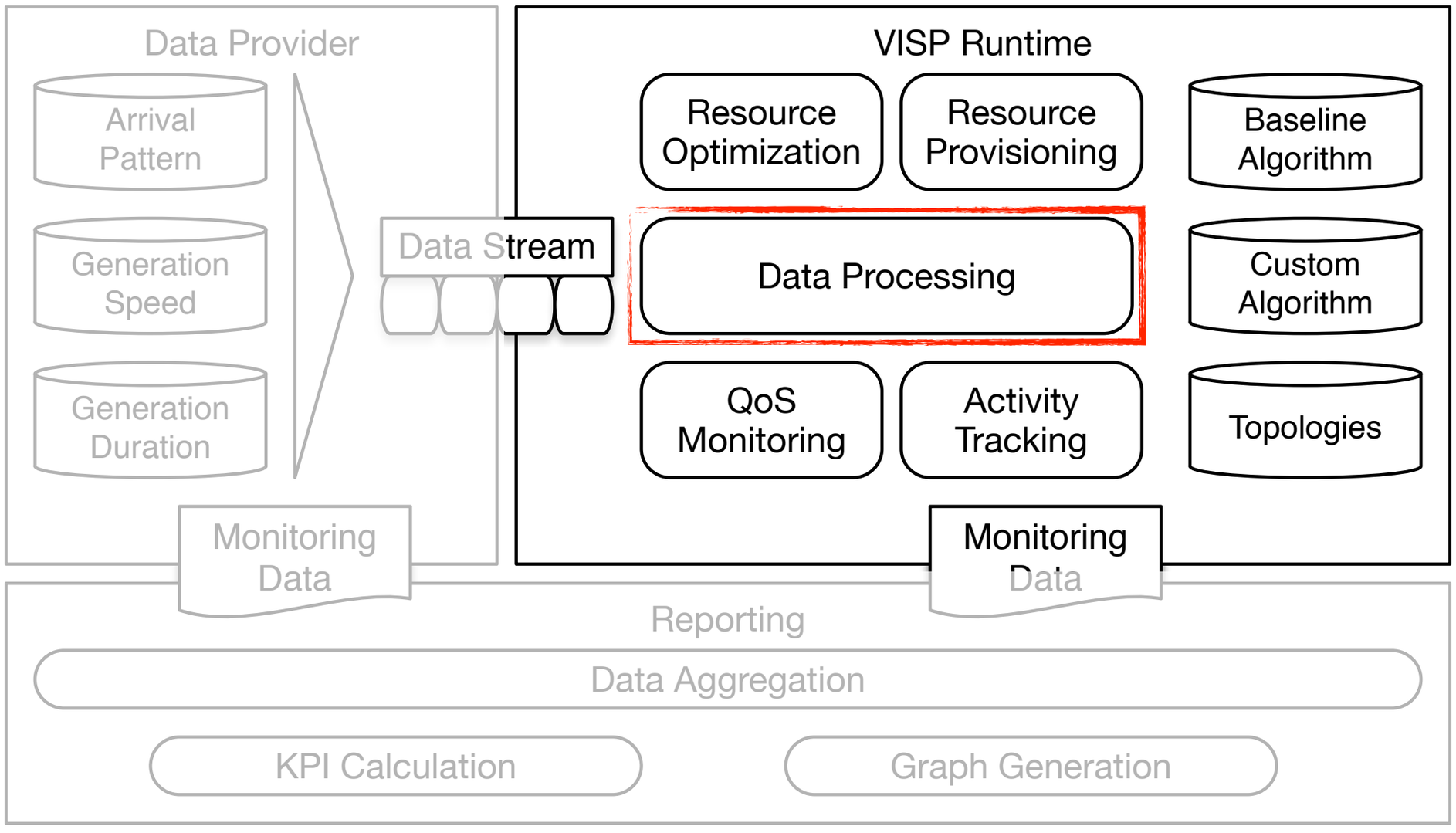
VISP Runtime



Data Provider
Arrival Pattern
Generation Speed
Generation Duration
Monitoring Data
VISP Runtime
Resource Optimization
Resource Provisioning
Baseline Algorithm
Data Stream
Data Processing
Custom Algorithm
QoS Monitoring
Activity Tracking
Topologies
Monitoring Data
Reporting
Data Aggregation
KPI Calculation
Graph Generation

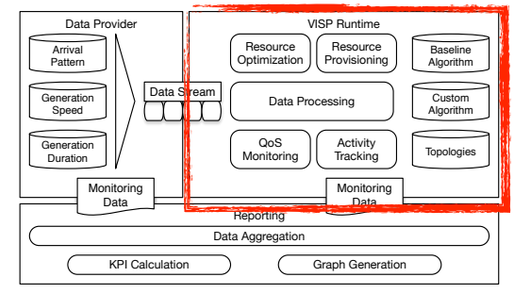
VISP Runtime

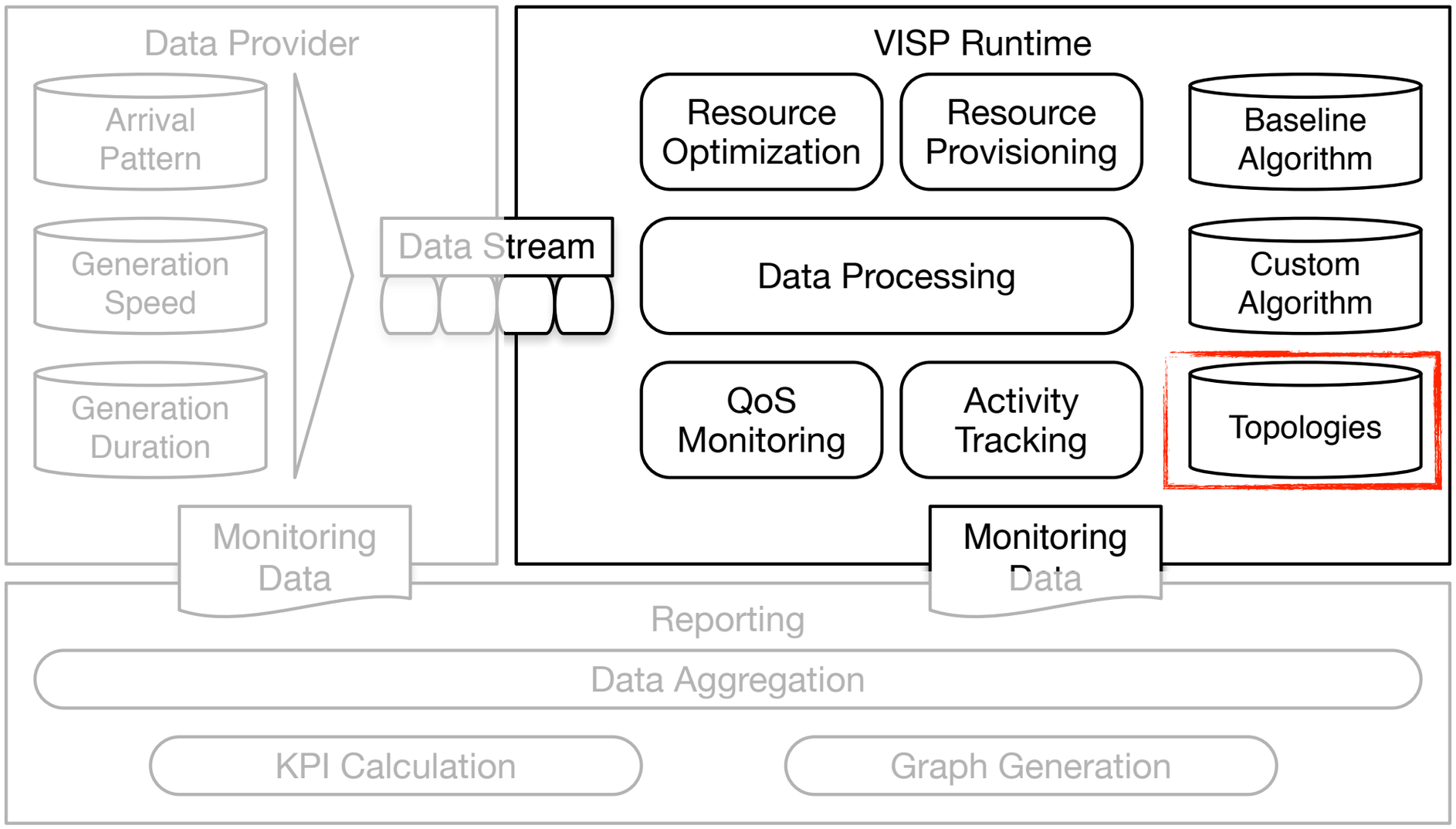




Process streaming data

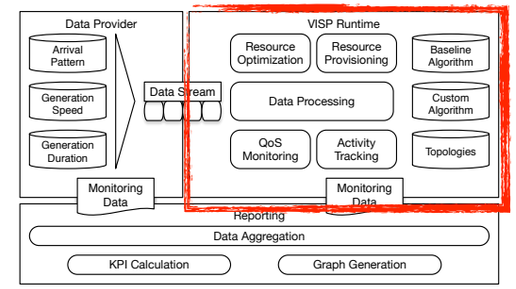
VISP Runtime

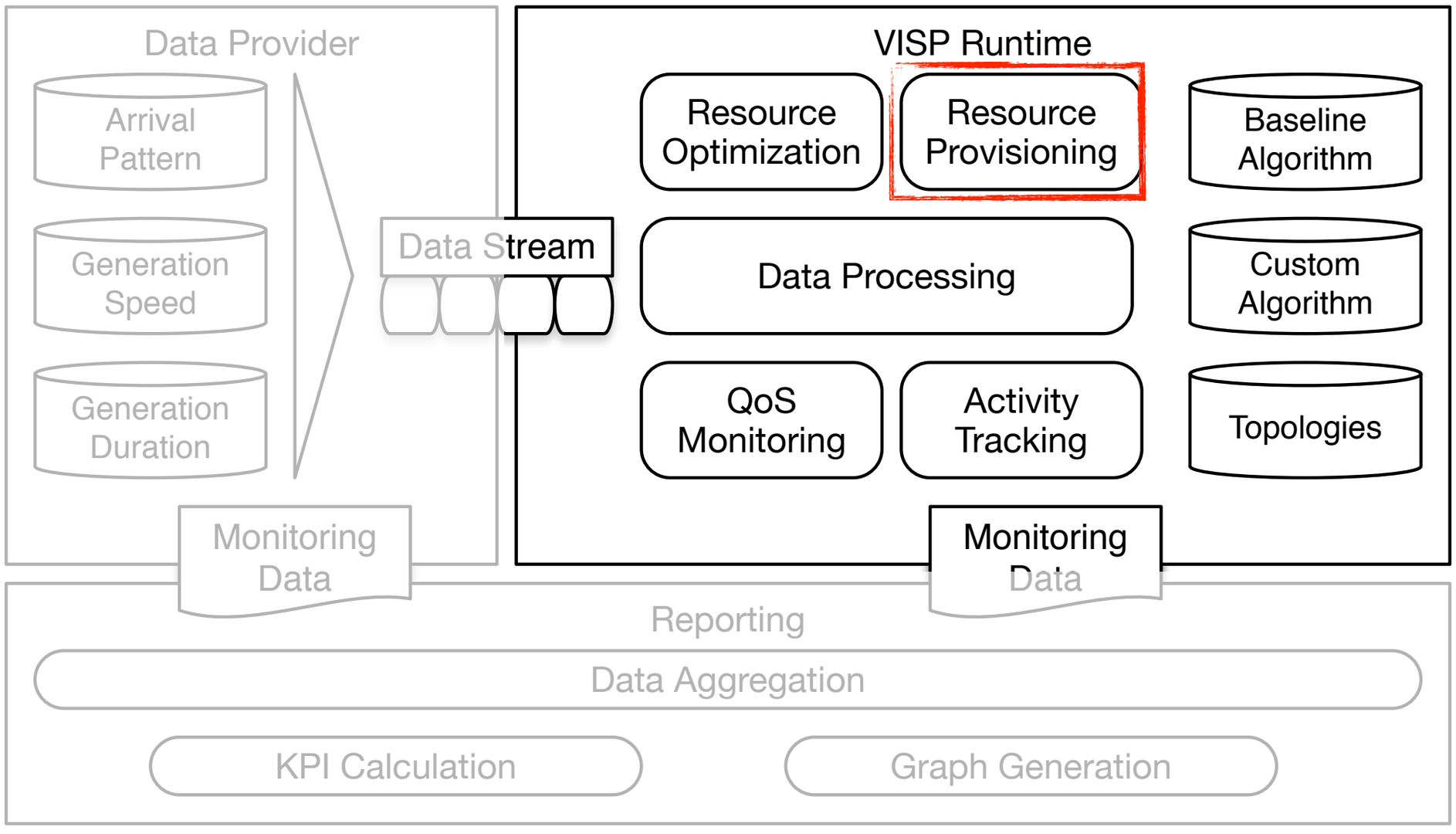




Route streaming data

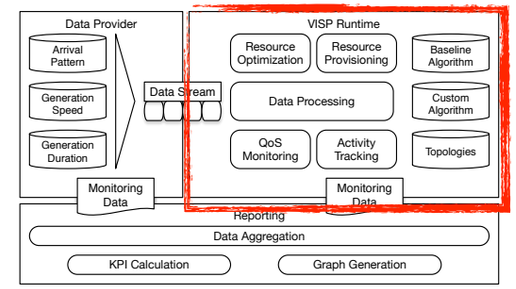
VISP Runtime

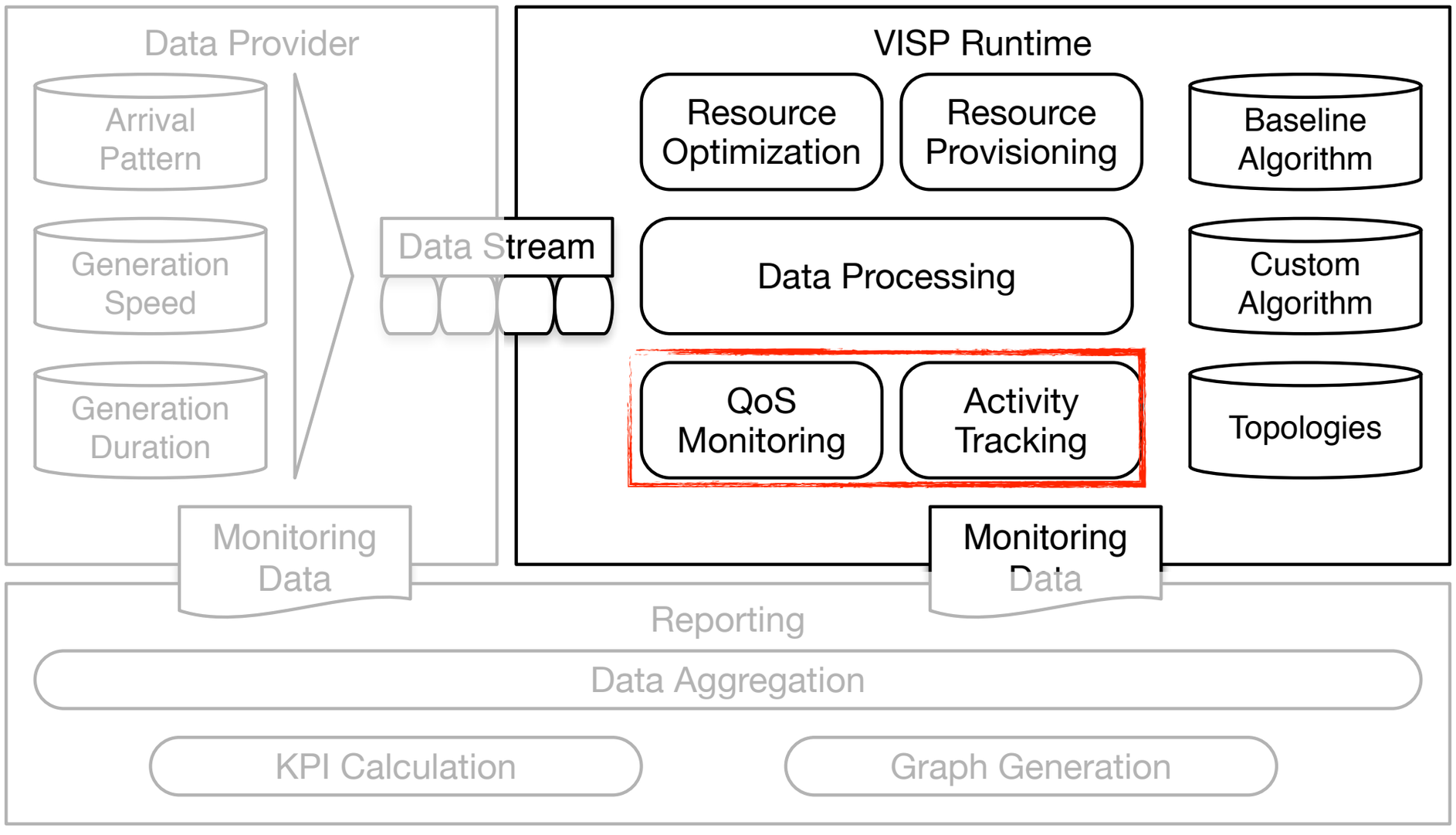




Provision computational resources

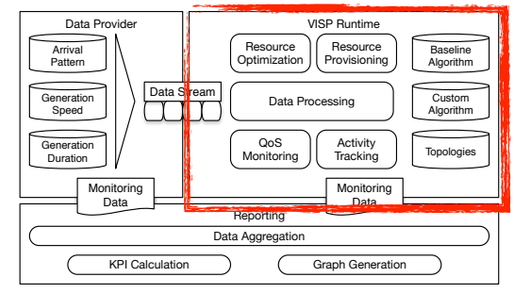
VISP Runtime

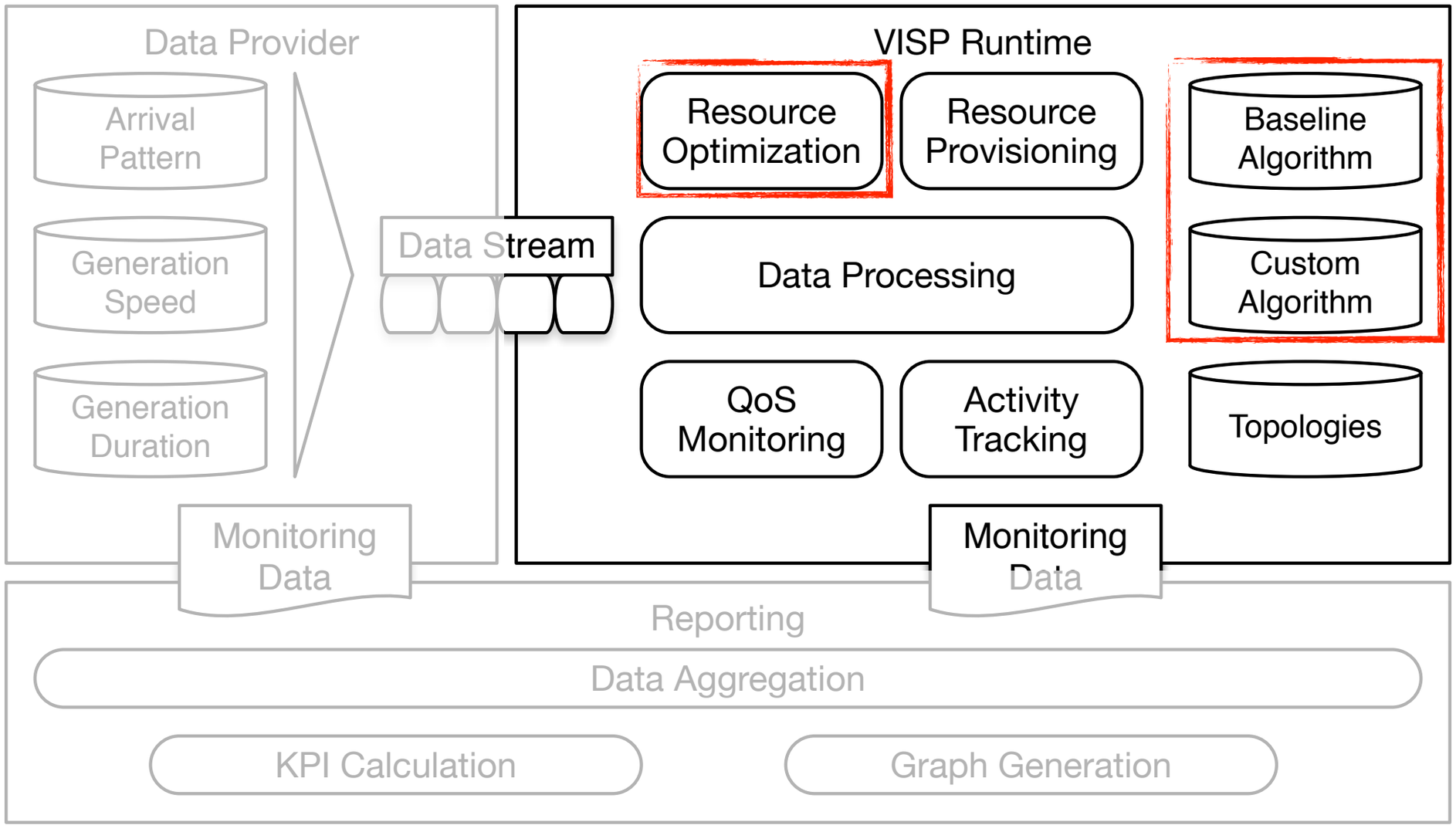




Monitor activities

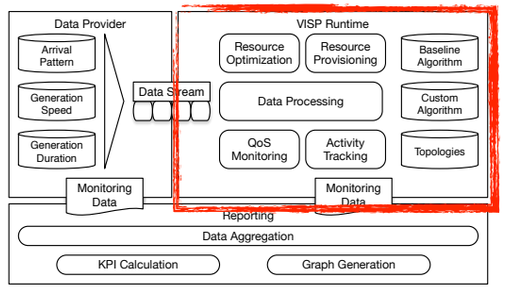
VISP Runtime





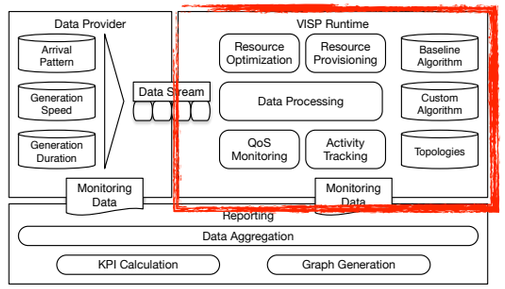
Optimize resource provisioning

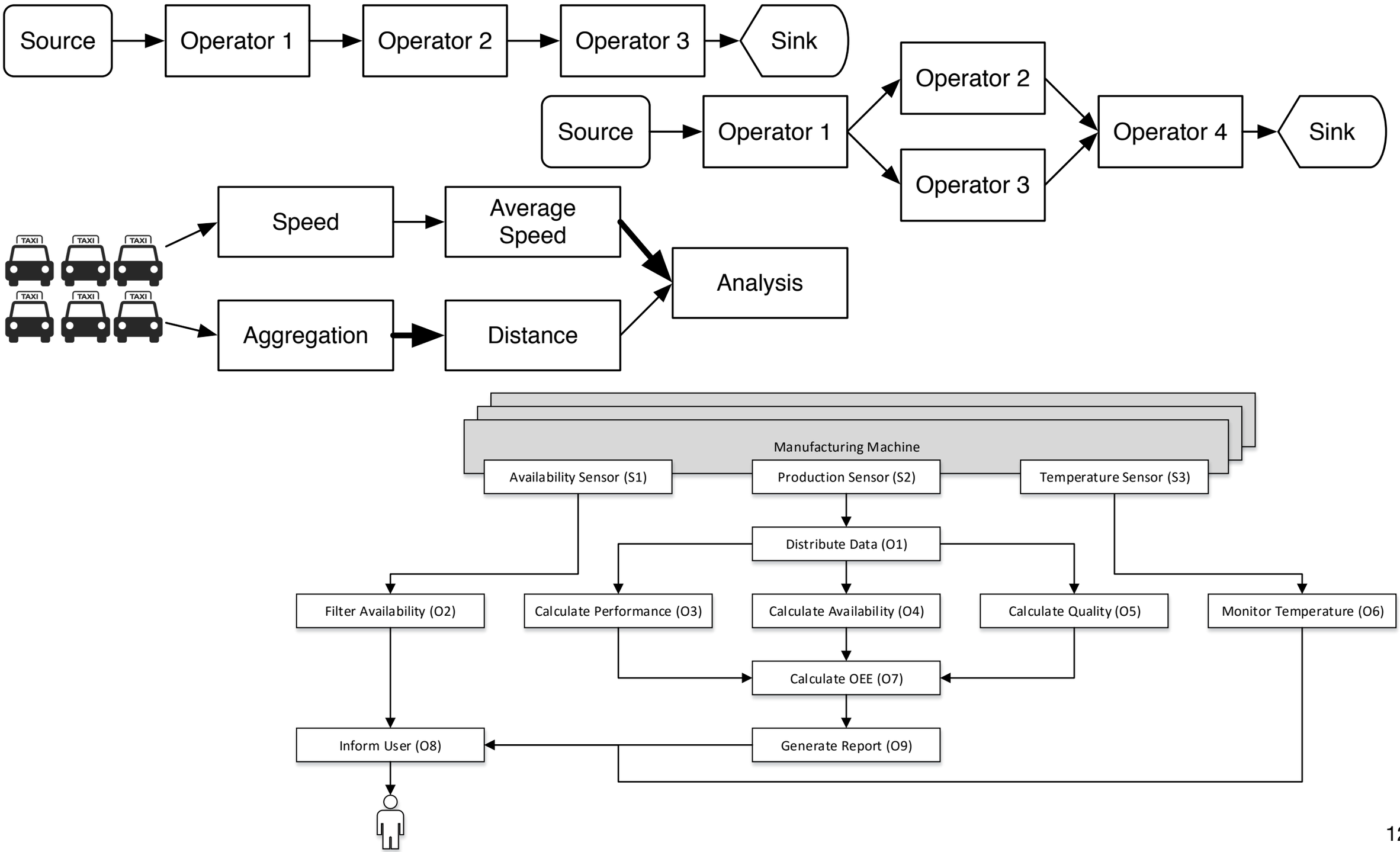
VISP Runtime - Base Line Algorithms

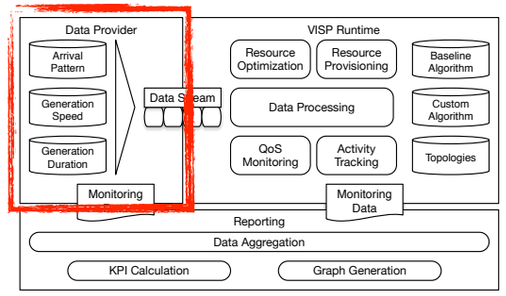


- Fixed Provisioning
  - Over Provisioning
  - Under Provisioning
- Threshold Based Provisioning
  - CPU
  - Memory
  - Queue Size

VISP Runtime - Topologies





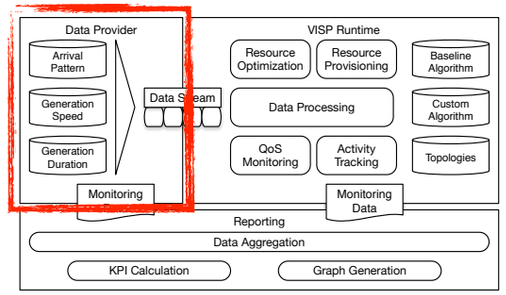


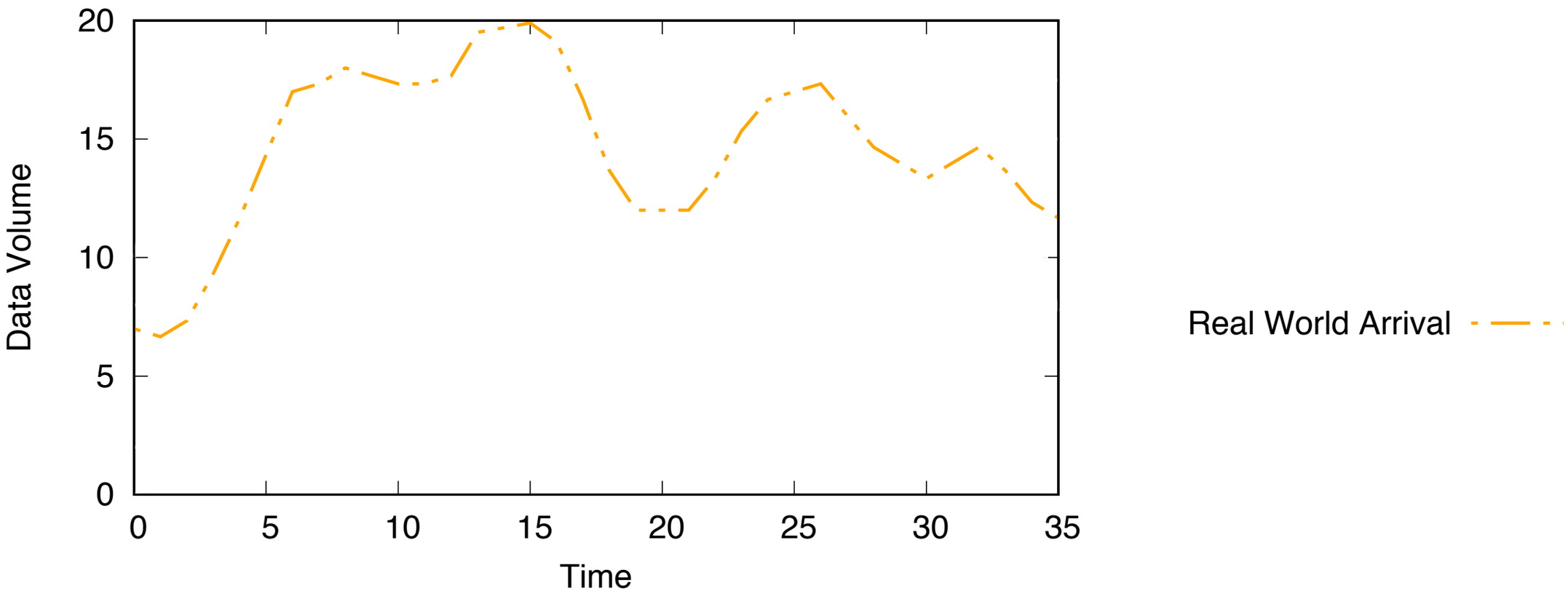
Data Provider

Replay data streams with

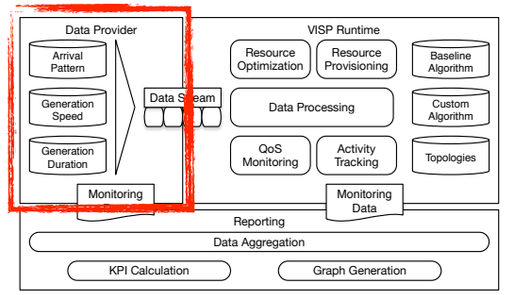
- predefined **arrival pattern** at
- predefined **generation velocities** with
- predefined **volumes** and
- predefined **message structures**.

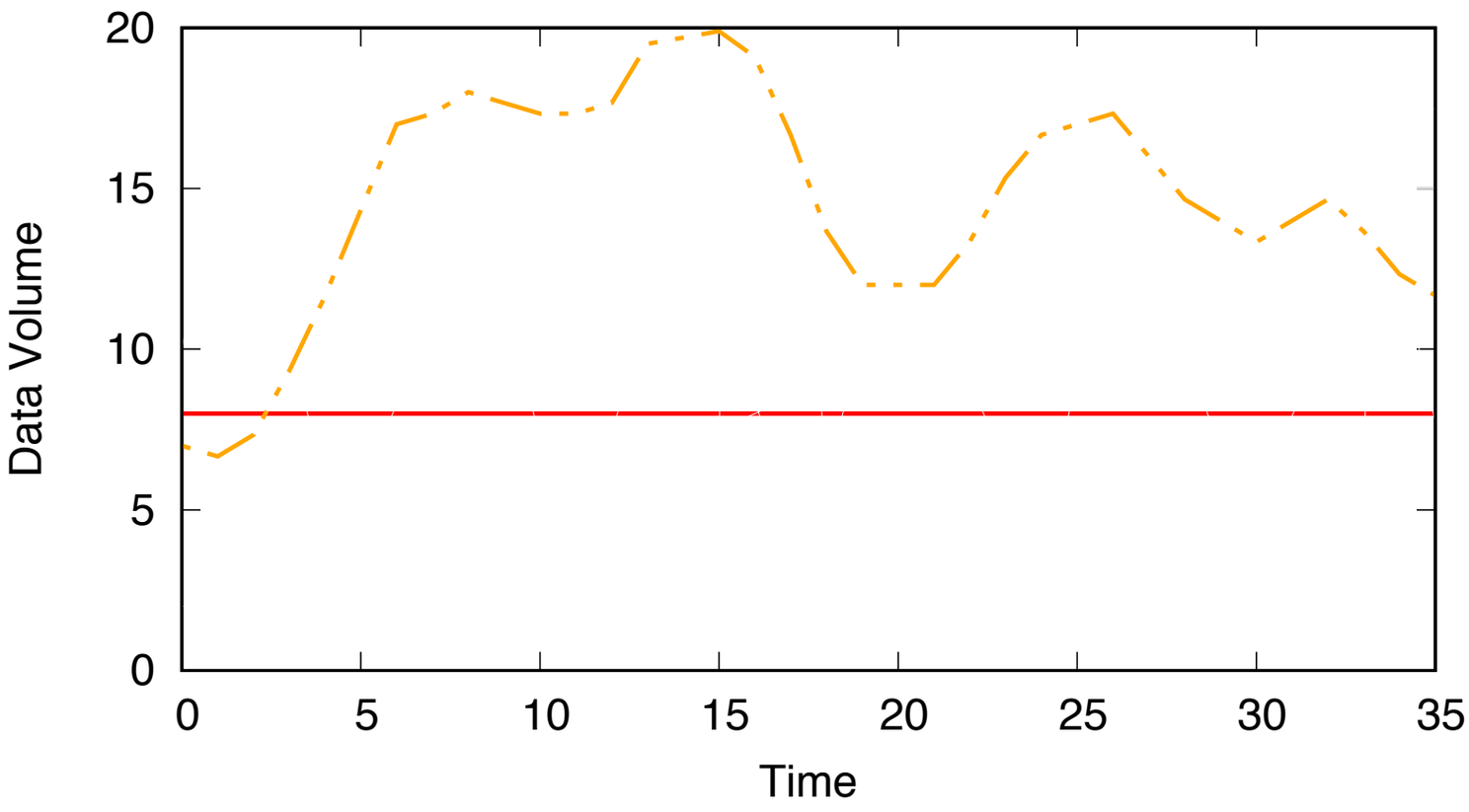
Data Provider - Arrival Pattern



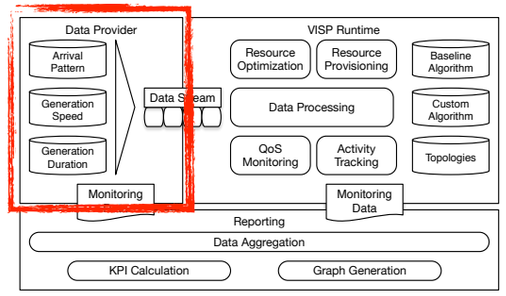


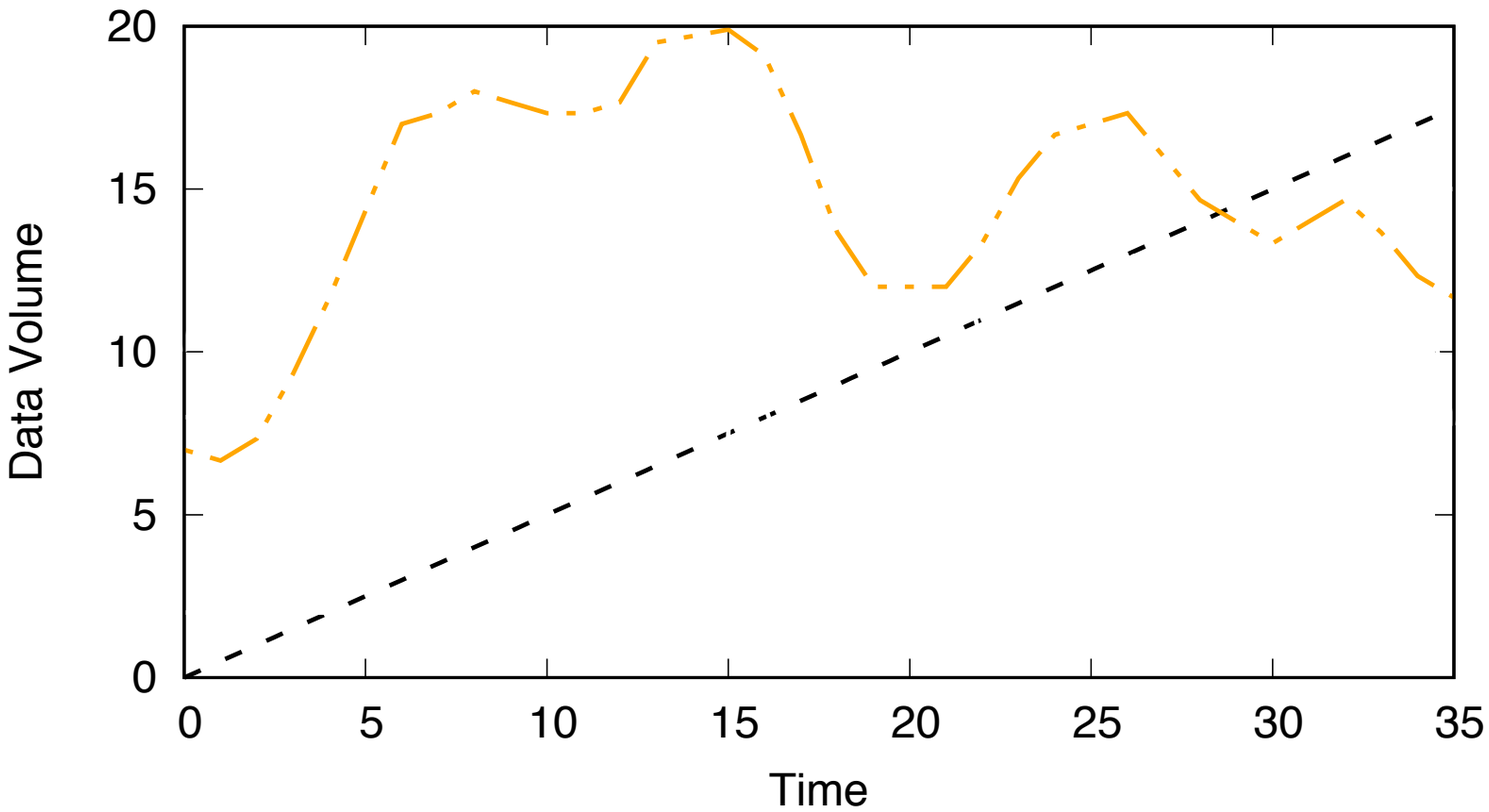
Data Provider - Arrival Pattern



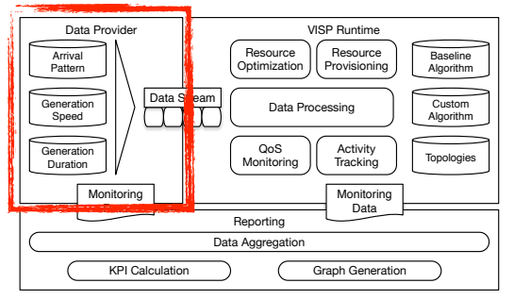


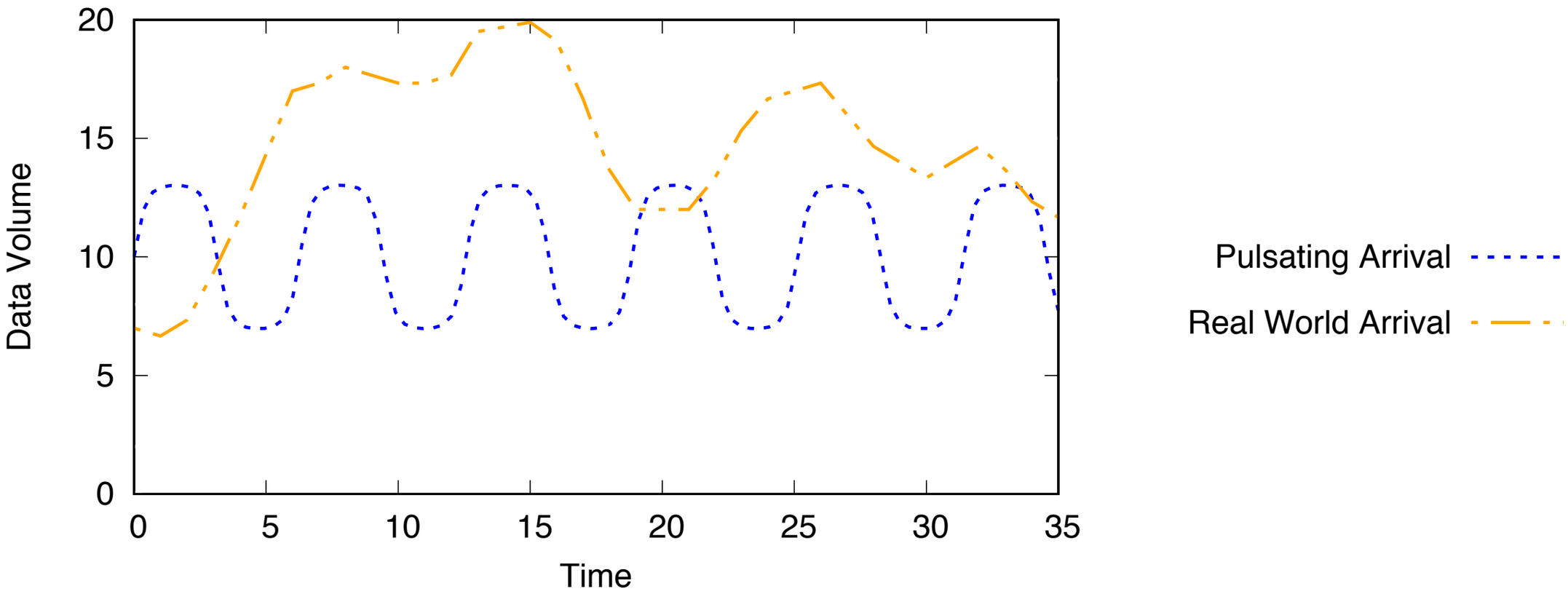
Data Provider - Arrival Pattern



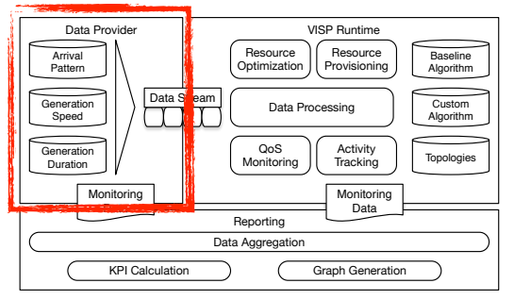


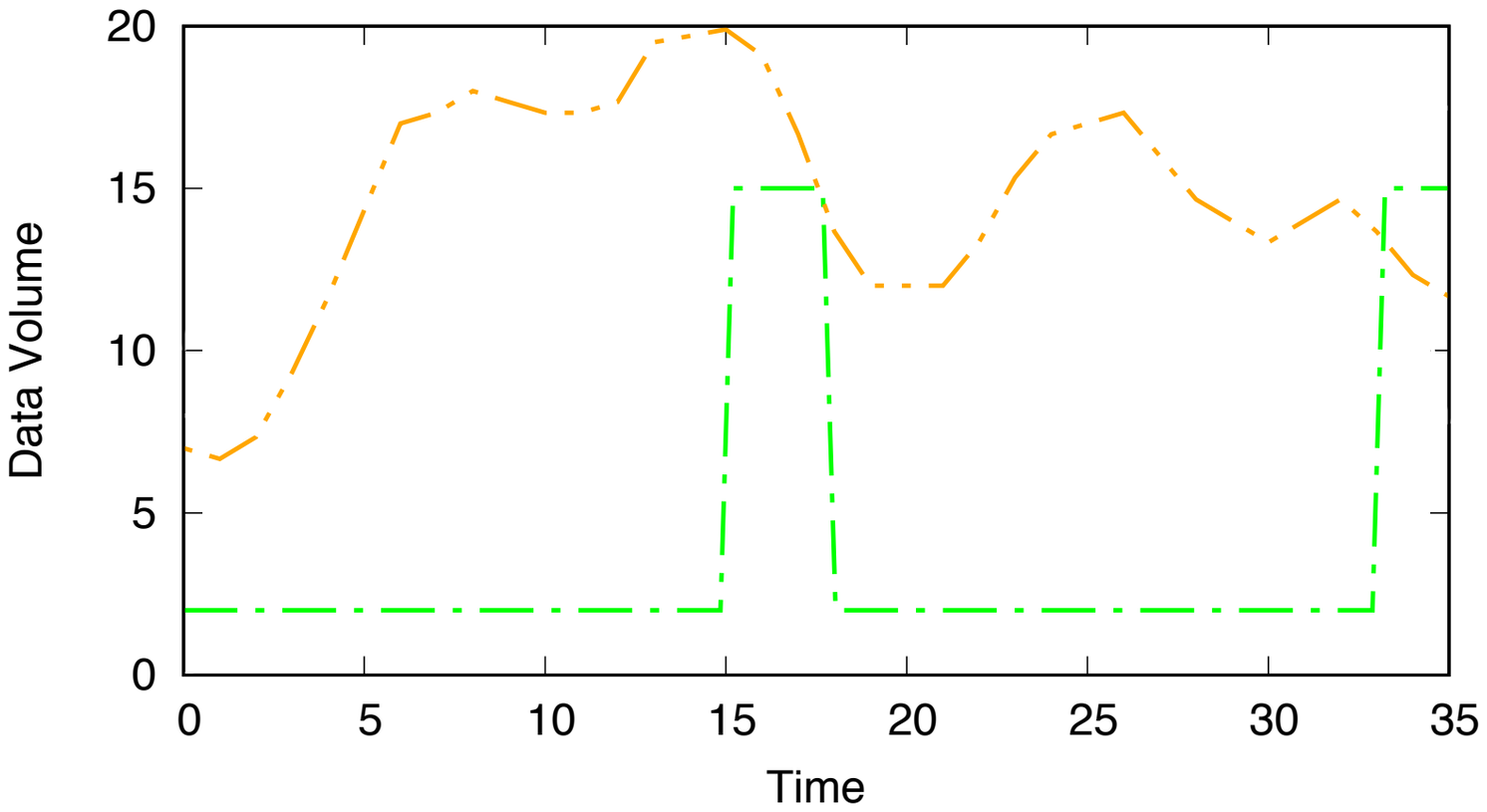
Data Provider - Arrival Pattern

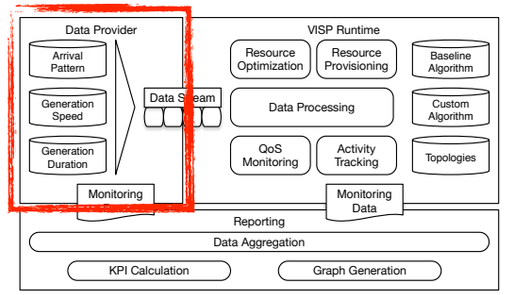




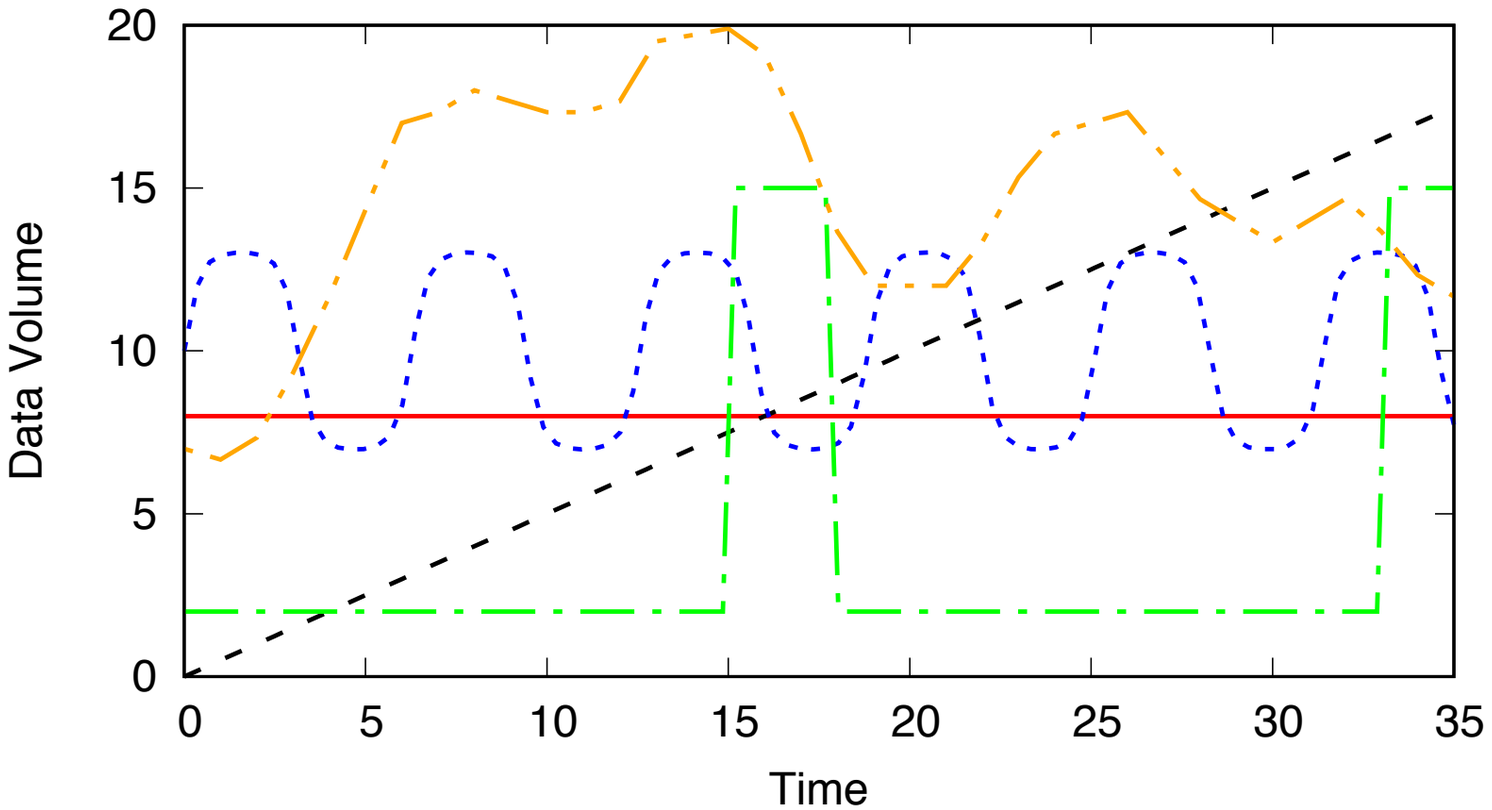
Data Provider - Arrival Pattern



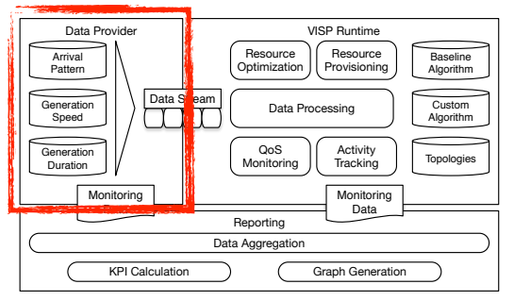




Data Provider - Arrival Pattern

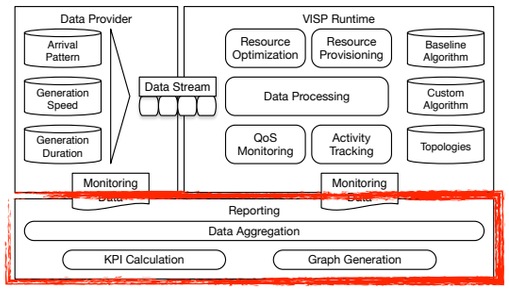


Data Provider - Message Structures



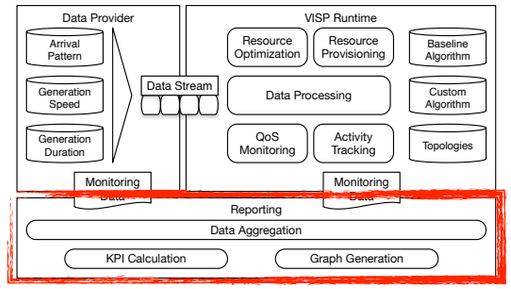
- Simple messages to trigger data processing events
- GPS time-series from taxis
- Sensor data from manufacturing machines

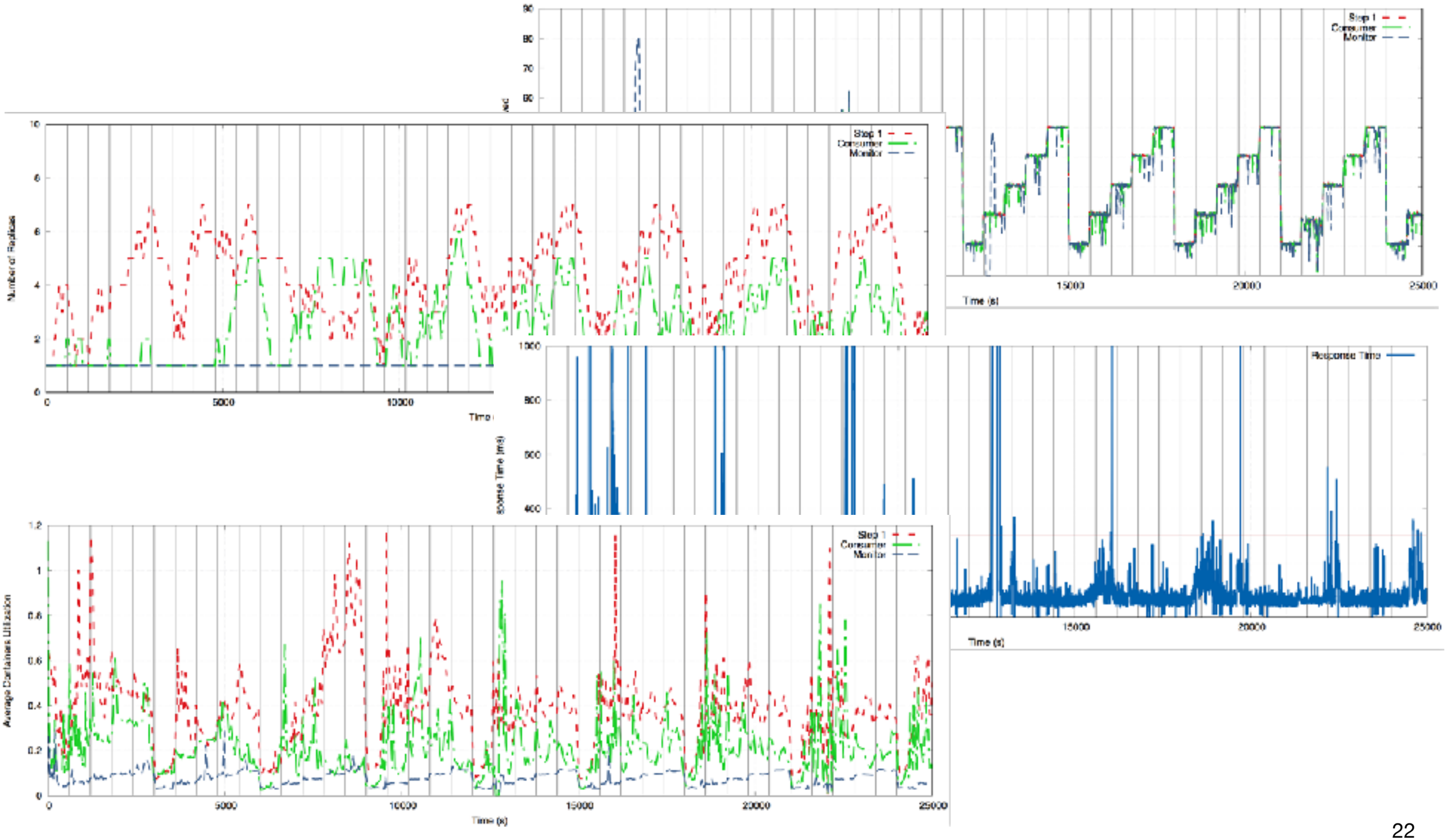
Reporting



- Aggregate information from monitoring data
- Generate KPIs
- Generate figures

Reporting


Data Provider
Arrival Pattern
Generation Speed
Generation Duration
Monitoring
Data Stream
VISP Runtime
Resource Optimization
Resource Provisioning
Baseline Algorithm
Data Processing
Custom Algorithm
QoS Monitoring
Activity Tracking
Topologies
Monitoring
Reporting
Data Aggregation
KPI Calculation
Graph Generation


Number of Replicas
Time (s)
Step 1
Consumer
Monitor
Response Time (ms)
Response Time
Average Container Utilization
Time (s)

Conclusion

Conclusion

VISP Testbed provides a toolkit

- to replay data streams and reproduce evaluation results
- to benchmark custom resource provisioning algorithm against existing ones

Outlook

Outlook

- Increase benchmark algorithm library
- Increase topology library and stream processing operator library
- Integrate topologies and data from the DEBS Grand Challenges
- Provide the VISP Testbed as a Service

Q & A

Christoph Hochreiner
c.hochreiner@infosys.tuwien.ac.at
https://github.com/visp-streaming/