OVERVIEW OF THE STRYMON PROJECT

John Liagouris
liagos@inf.ethz.ch

REAL-TIME DATACENTER ANALYTICS

eBay, San Jose

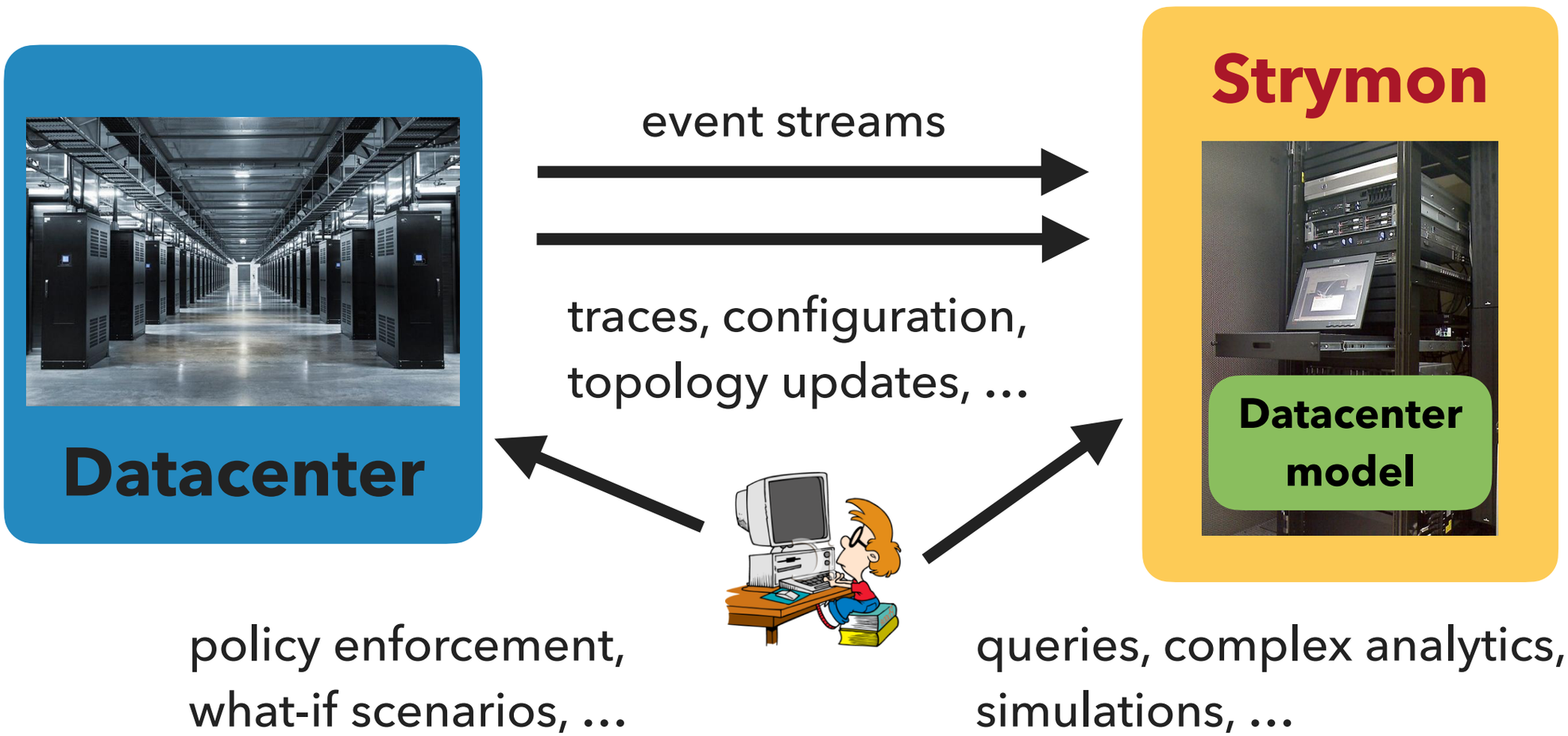
21 February 2019

ABOUT ME

- Senior researcher at ETH Zürich
  - Systems Group: https://www.systems.ethz.ch/
- Visiting scholar at UC Berkeley
  - RISELab: https://rise.cs.berkeley.edu
- Research interests
  - Dataflow systems
  - Real-time data analytics
  - Databases
- Current projects
  - Strymon: http://strymon.systems.ethz.ch
  - Ray: https://rise.cs.berkeley.edu/projects/ray/

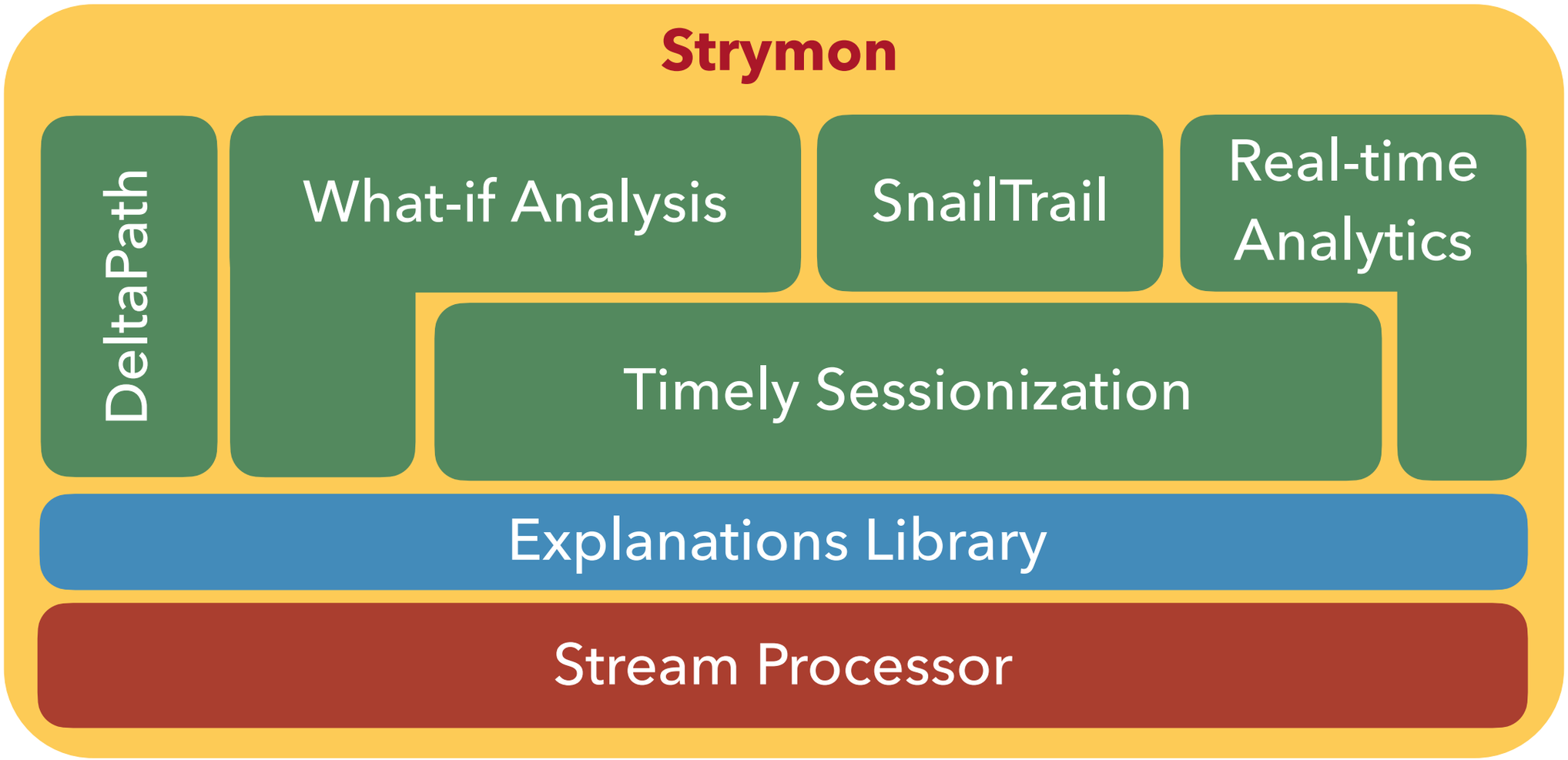
DATACENTER MANAGEMENT

THE BIG PICTURE: UNDERSTANDING THE DATACENTER

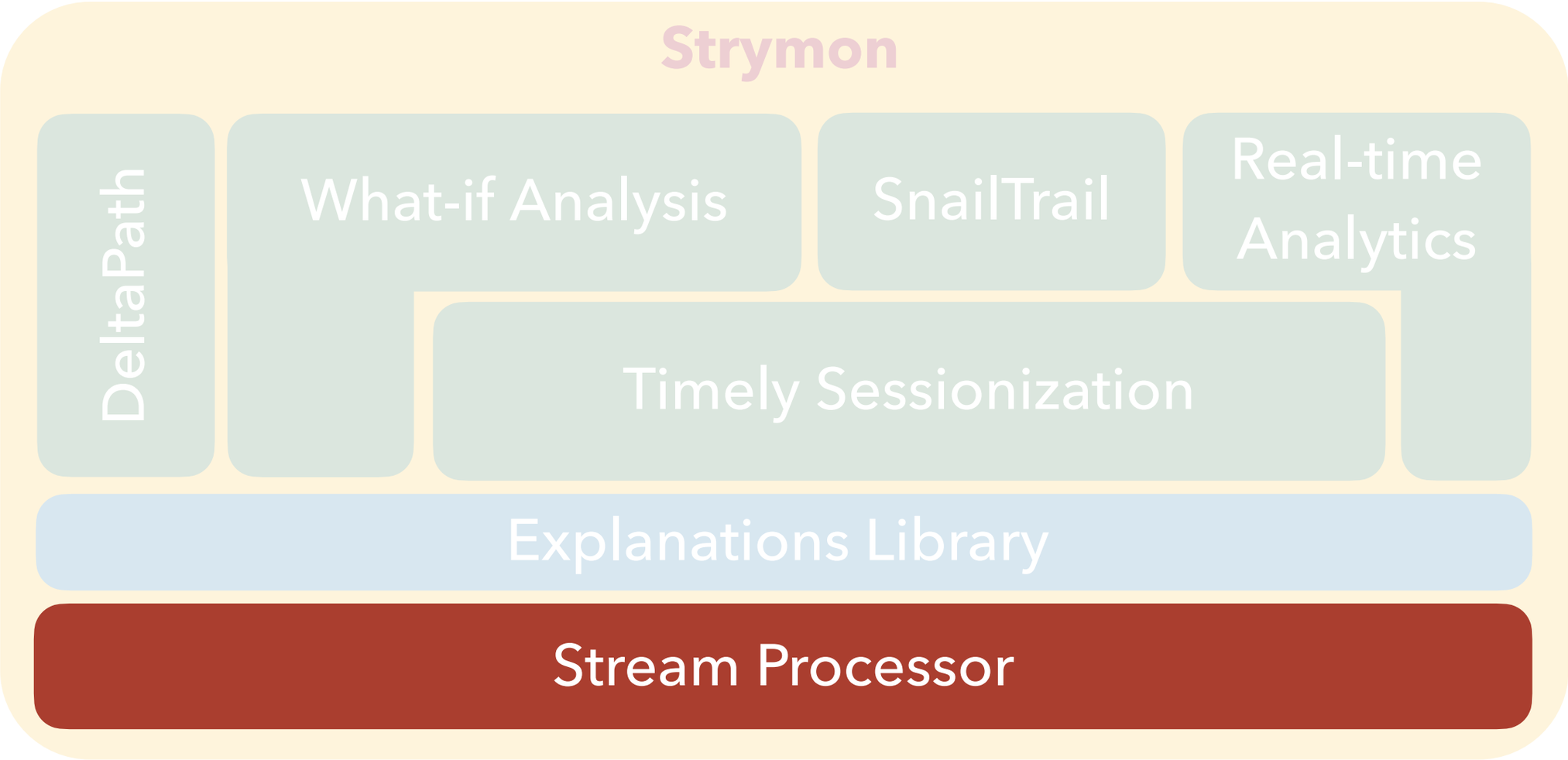


strymon.systems.ethz.ch

THE STRYMON STACK



THE STRYMON STACK



TIMELY DATAFLOW

D. Murray, F. McSherry, M. Isard, R. Isaacs, P. Barham, M. Abadi. Naiad: A Timely Dataflow System. In SOSP, 2013.

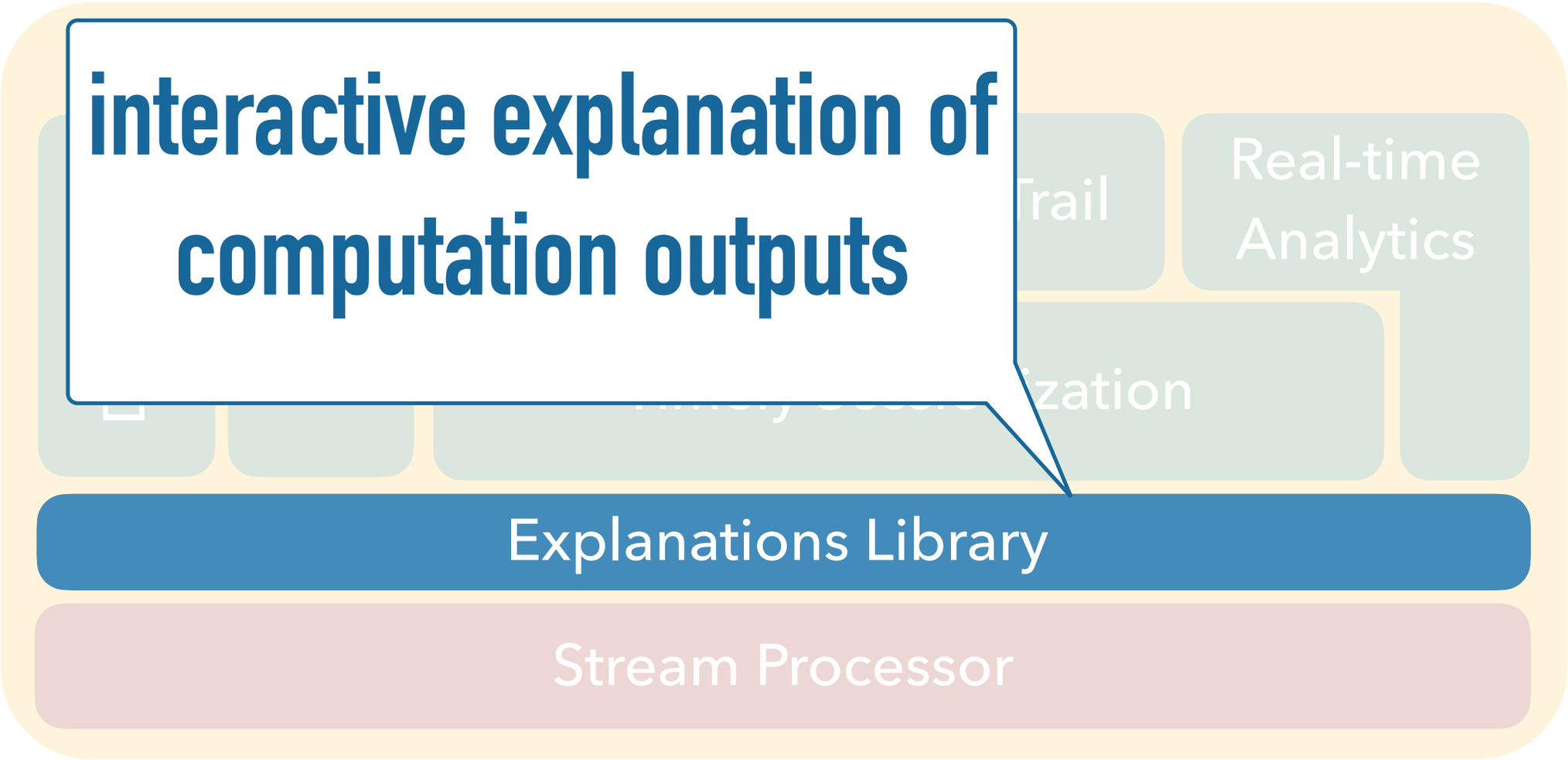
- A steaming engine for data-parallel computations
  - Cyclic dataflows
  - Logical timestamps (epochs)
  - Asynchronous execution
  - Low latency - High Throughput

DIFFERENTIAL DATAFLOW

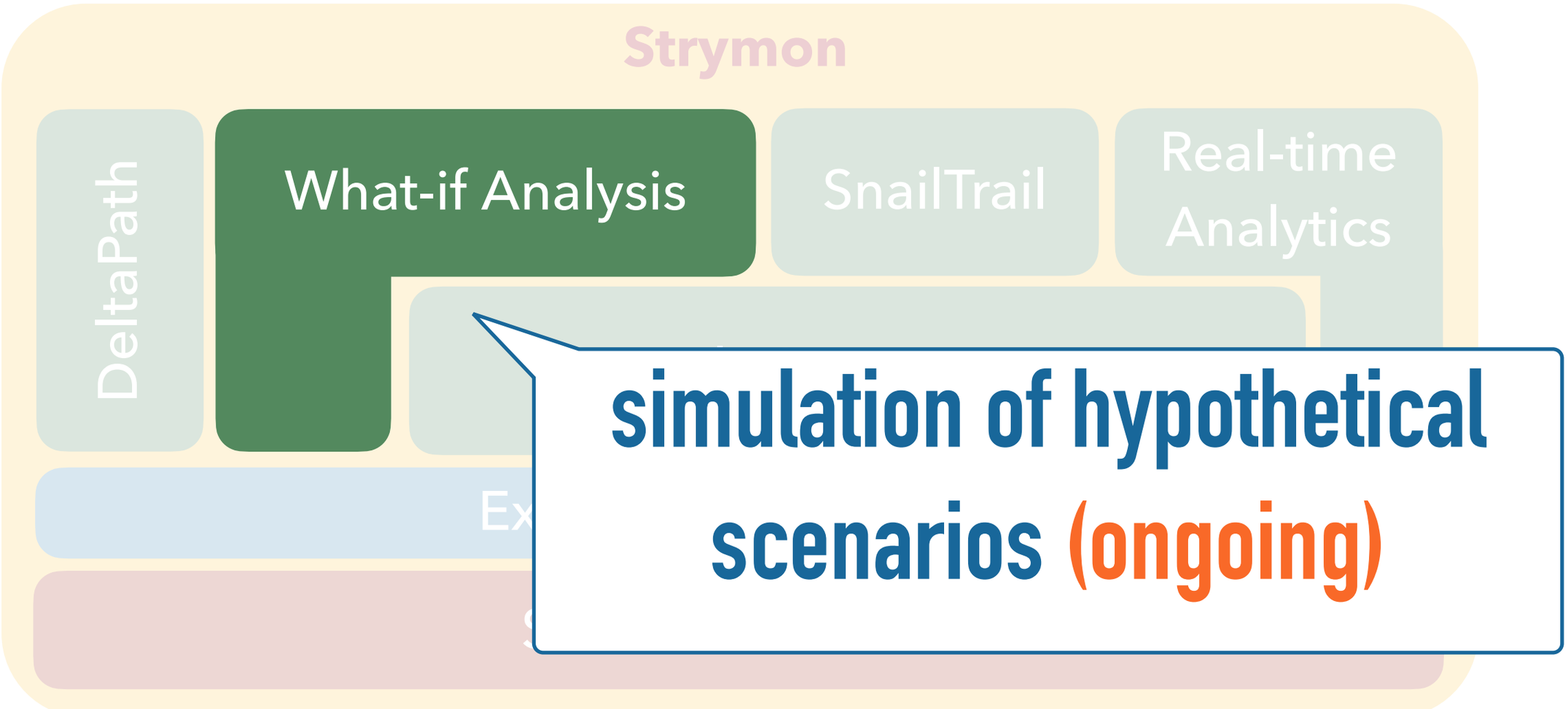
F. McSherry, D. Murray, R. Isaacs, M. Isard. *Differential Dataflow*. In CIDR, 2013.

- A library on top of Timely Dataflow for incremental computation

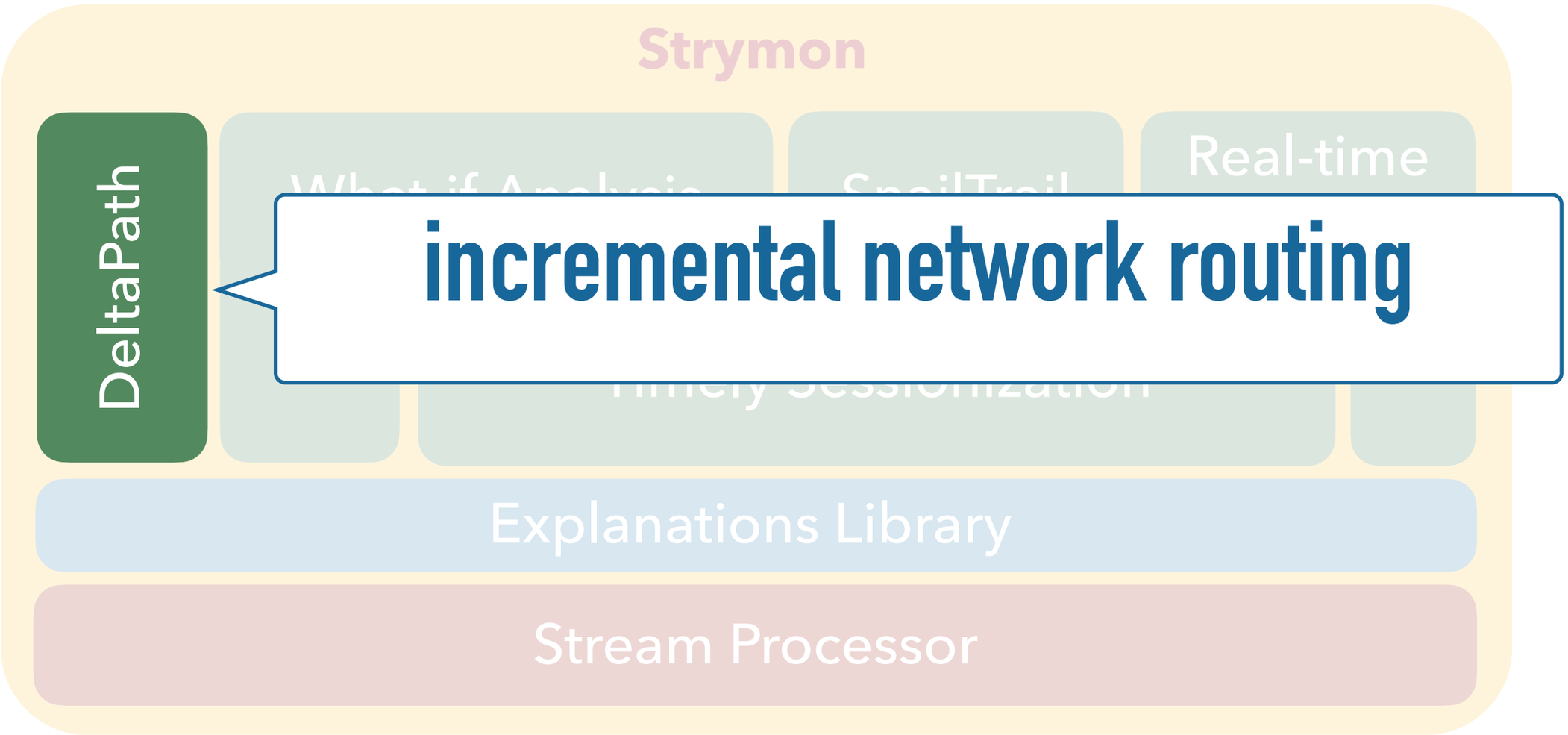
THE STRYMON STACK



THE STRYMON STACK

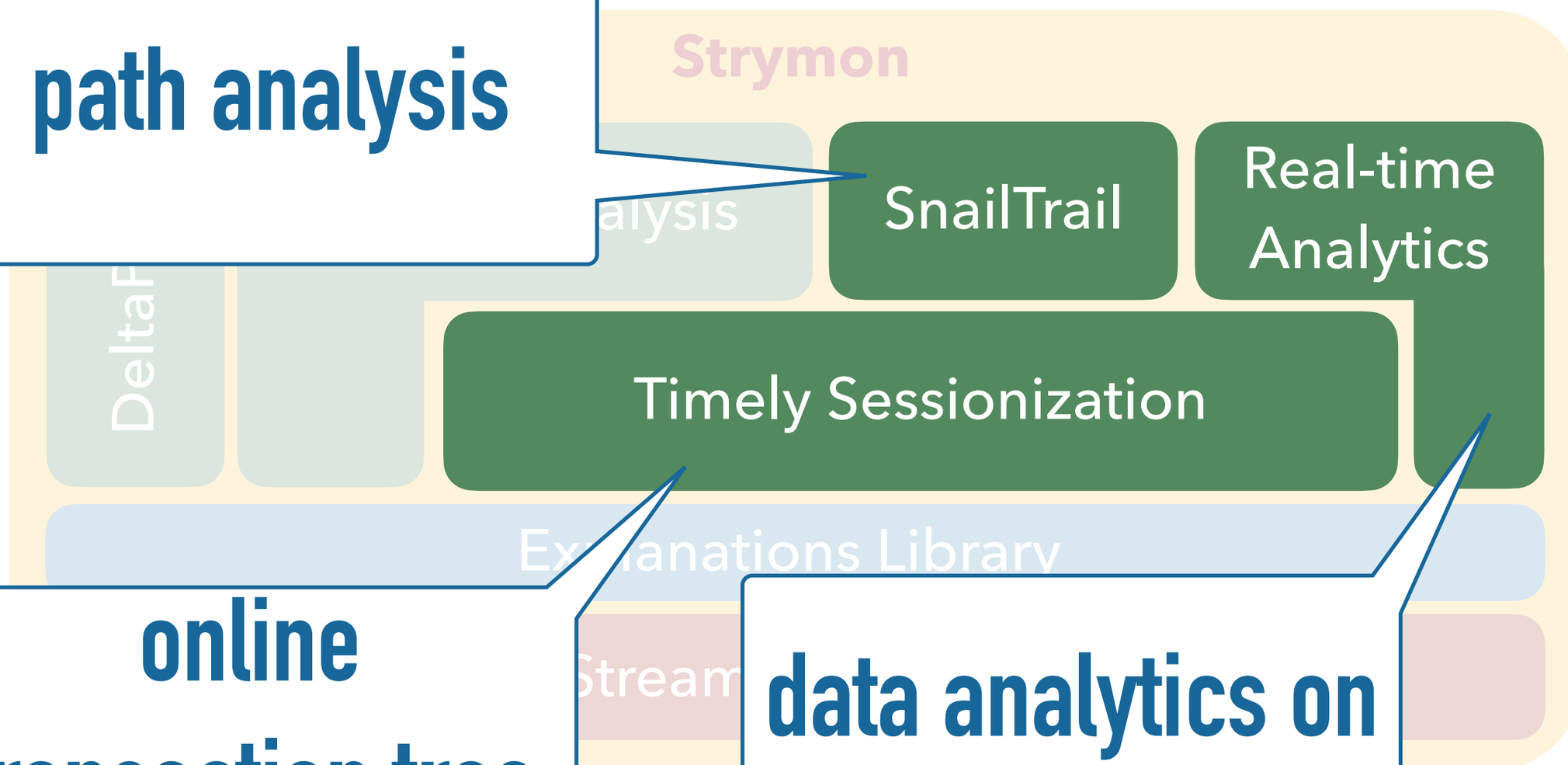


THE STRYMON STACK



FOCUS OF TODAY'S TALK

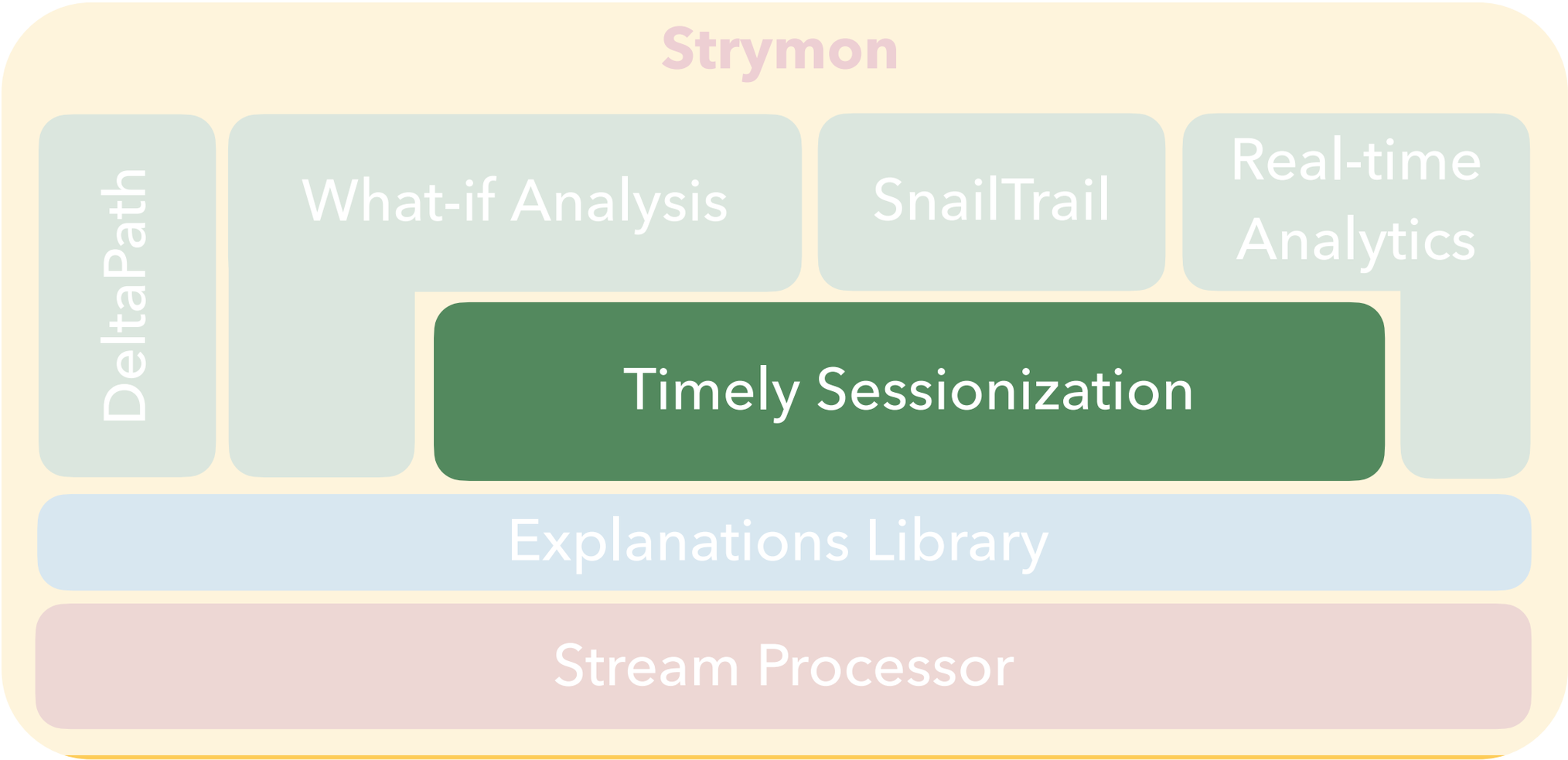
online critical path analysis



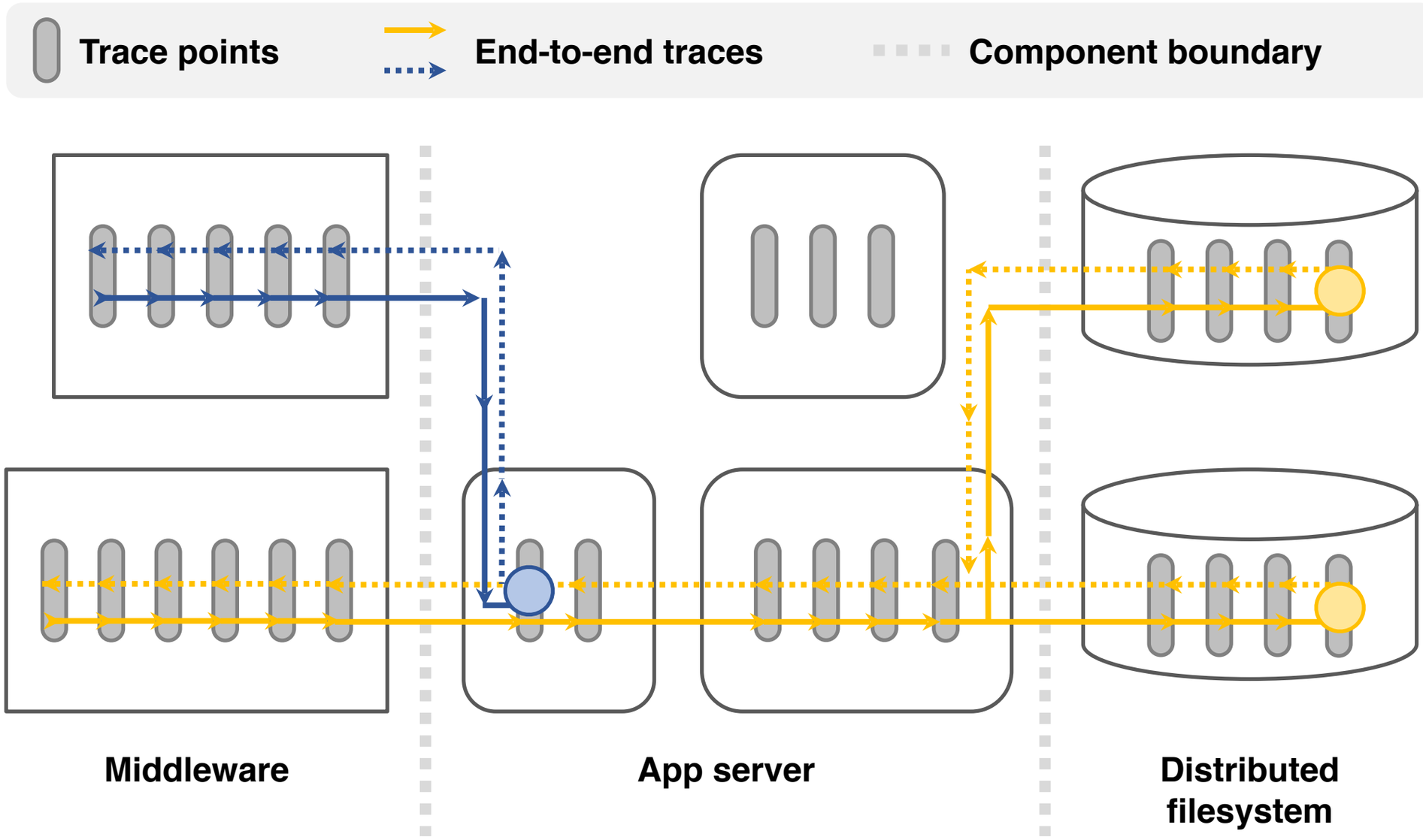
online transaction tree reconstruction

data analytics on streams of traces

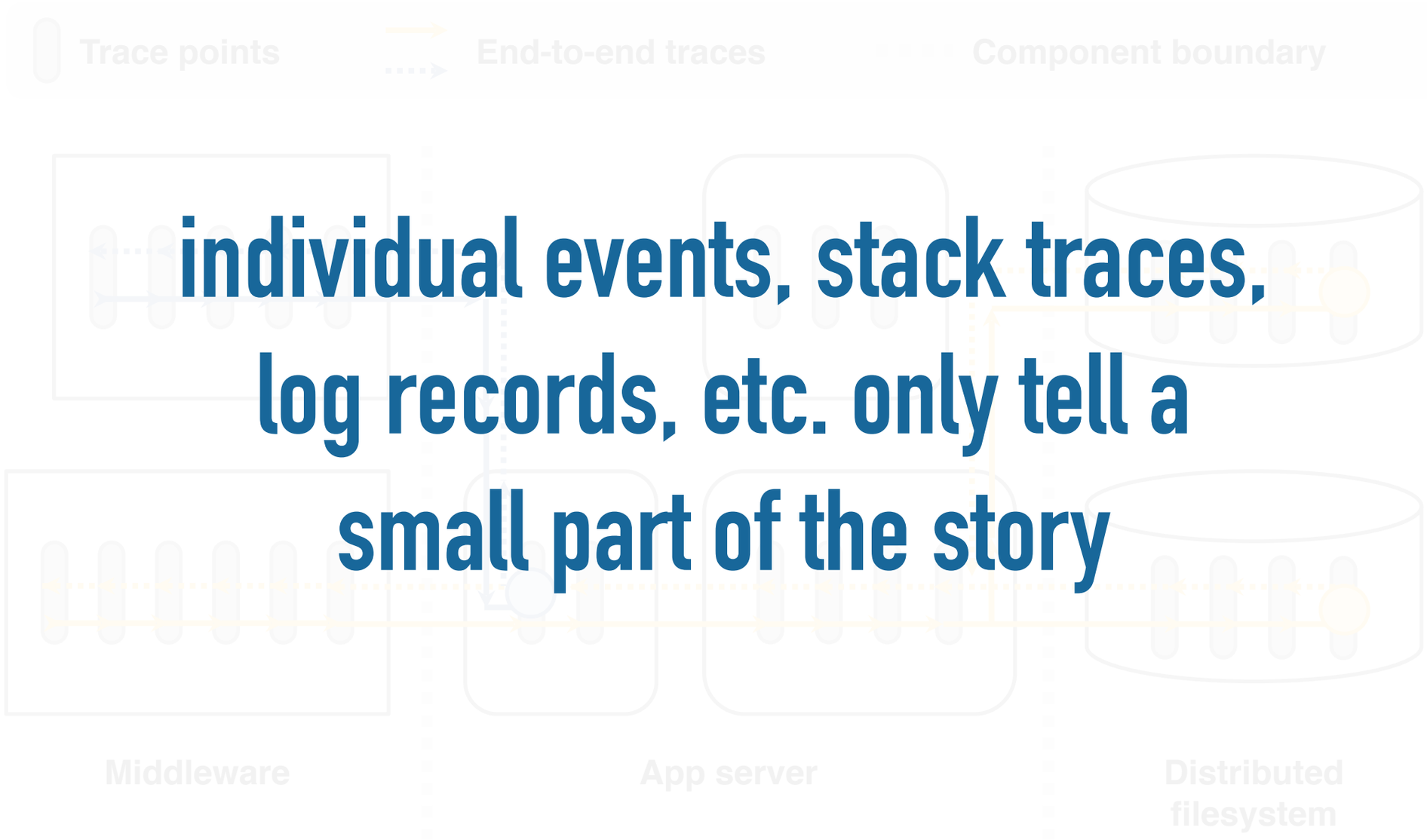
ONLINE SESSIONIZATION



DATACENTER STACK IS ALREADY HEAVILY INSTRUMENTED



DATACENTER STACK IS ALREADY HEAVILY INSTRUMENTED

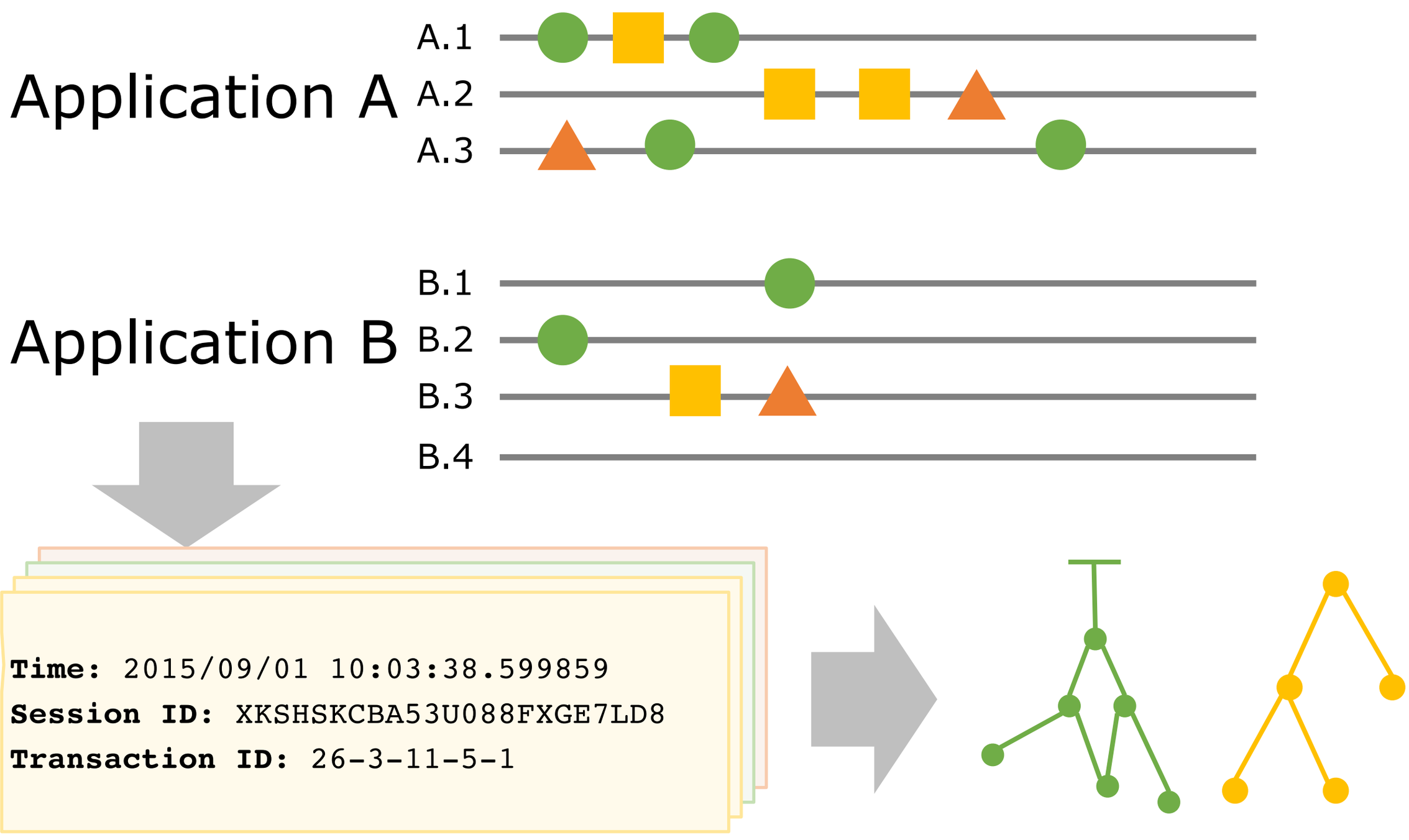


individual events, stack traces, log records, etc. only tell a small part of the story

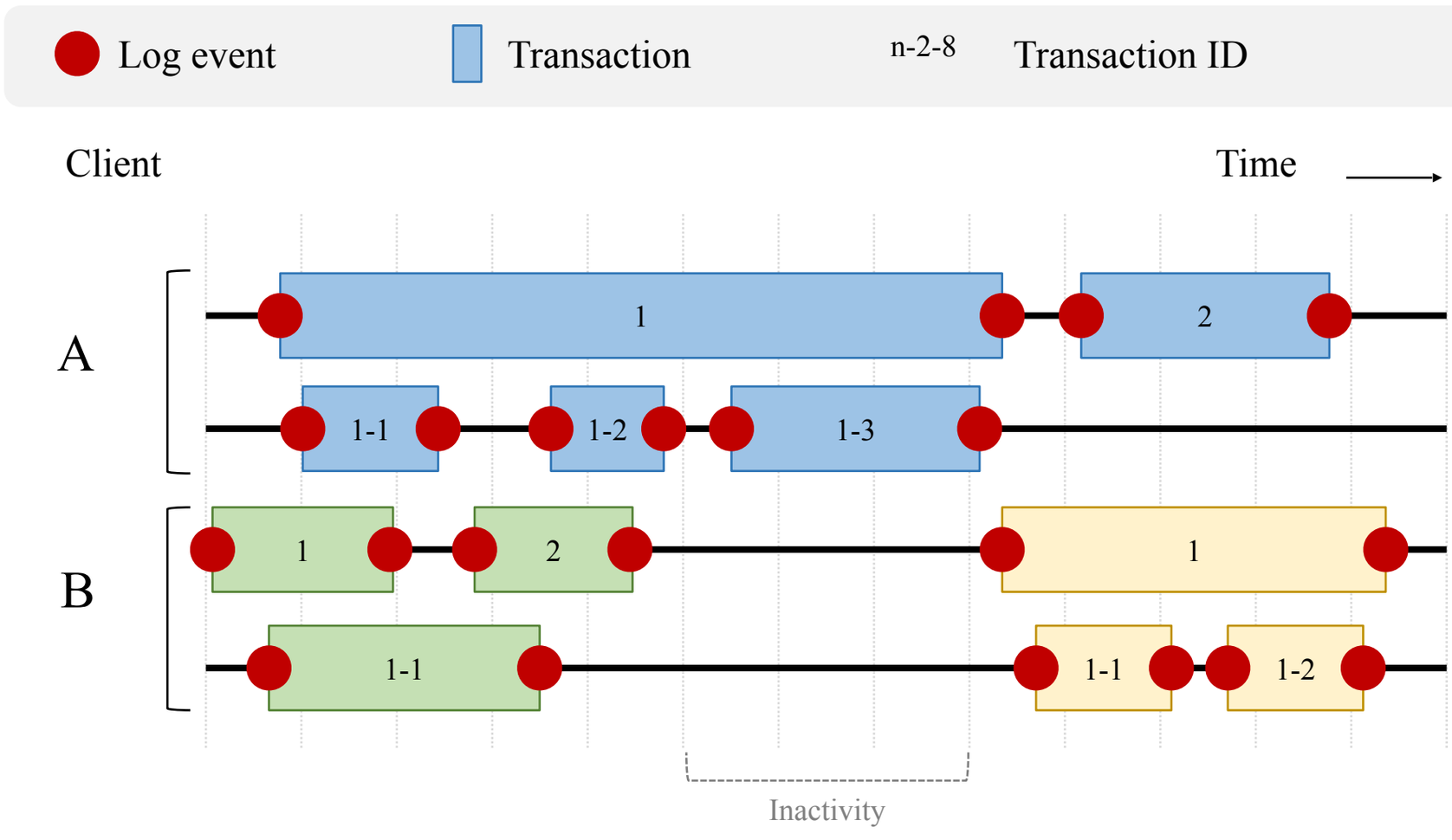
DATACENTER STACK IS ALREADY HEAVILY INSTRUMENTED

tracking lineage of individual events provides better insights

THE SESSIONIZATION PROBLEM

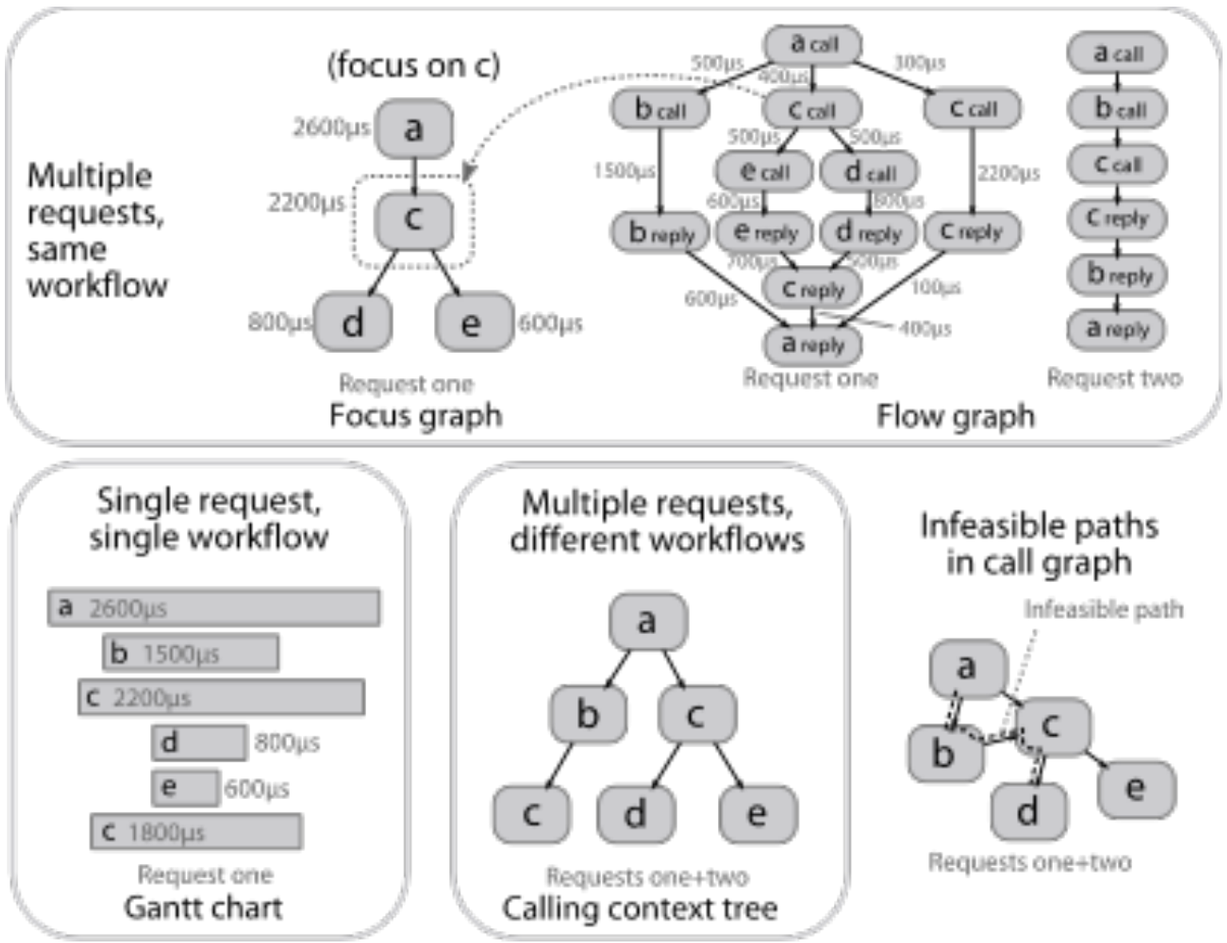


THE SESSIONIZATION PROBLEM



SESSION SUMMARIES FOR DATACENTER DIAGNOSTICS

Foundation for diagnostic, profiling, and monitoring tasks essential to the operation of the datacenter



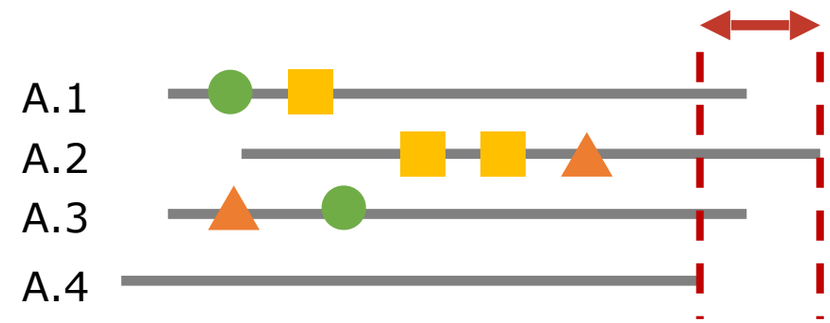
- User sessions
- Spans
- Call graphs
- Transaction trees
- Critical path
- Timing charts

Sambasivan, Raja R., et al. *So, you want to trace your distributed system? Technical Report, CMU-PDL-14-102*

CHALLENGES WHEN USING REAL LOGS

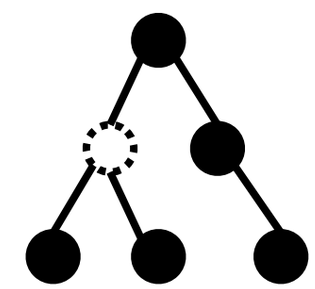
Out-of-order arrivals

Records arrive in non-deterministic order but within limited time frame (max. observed: 10 seconds)



Missing logs

33.7% have no session ID or transaction number
Incomplete or fragmented trees prevent dependency inference



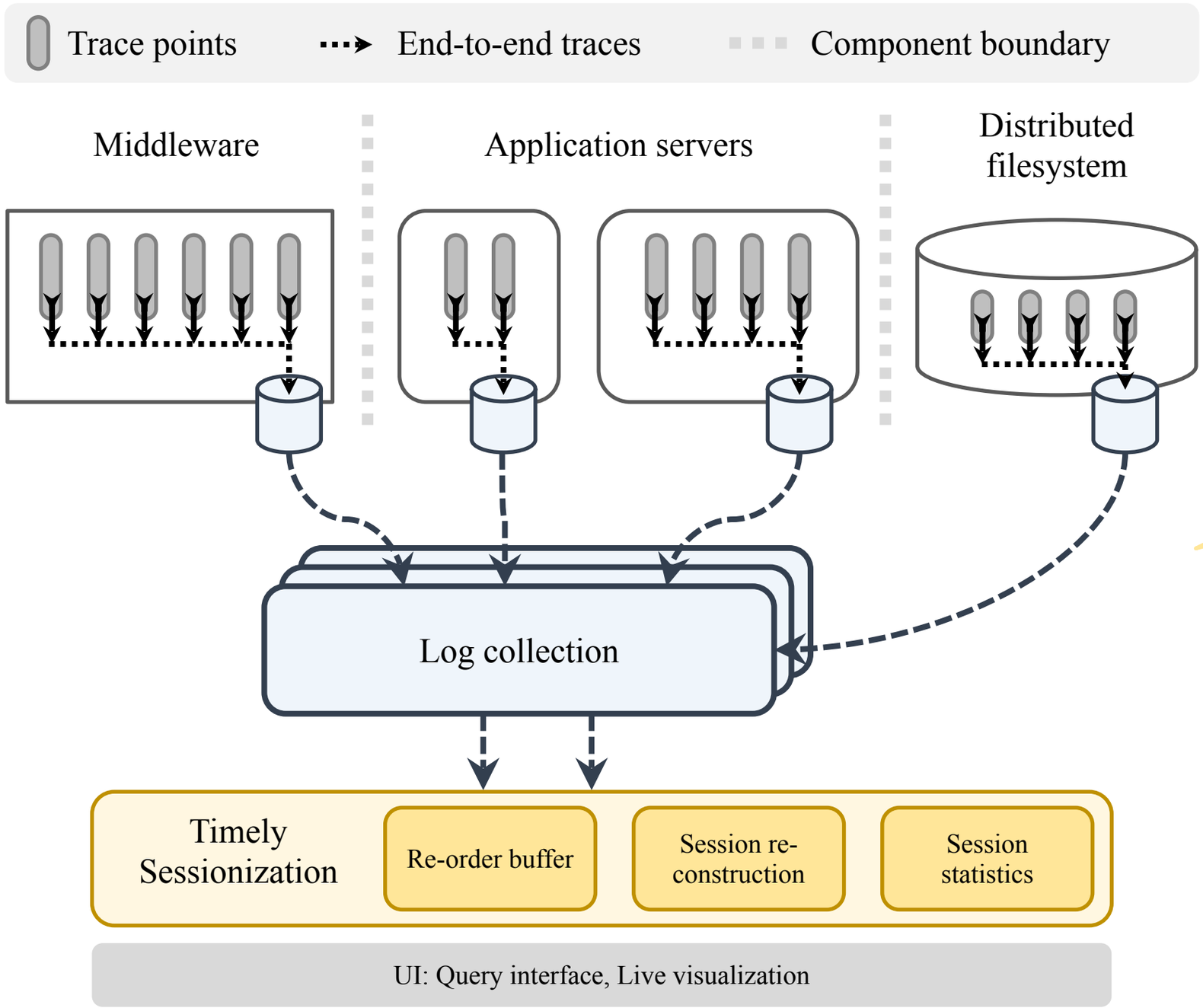
Very long sessions

Inherent skew, high memory requirements

Clock skew

Misordering: message appears to be received before sent
Trigger inversion: parent transaction starts after child

USE CASE: AMADEUS DATACENTER



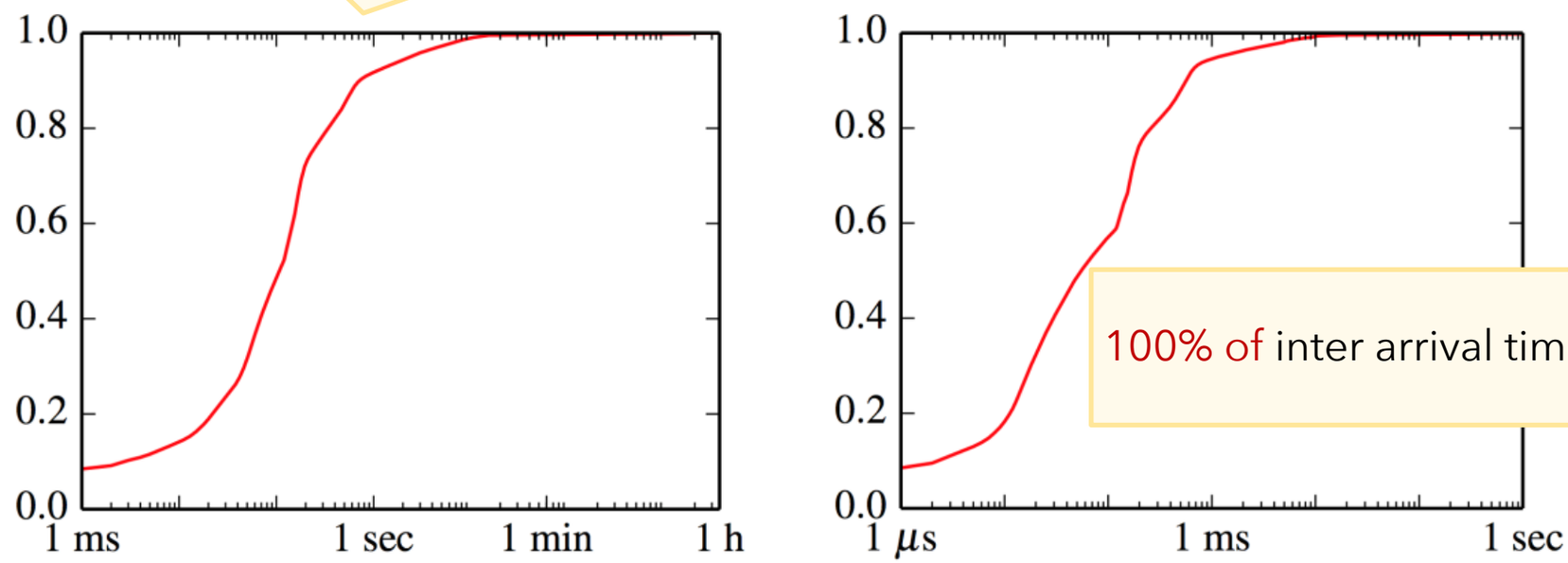
Logs spread across 1263 streams and 42 servers

Mean input rate: 1.3 million events/sec at 424.3 MB/sec

https://github.com/strymon-system/reconstruction

WORKLOAD CHARACTERISTICS

~95% of root transactions are short-lived with total time span of ≤ 2 seconds



(a) Transaction Tree Duration (b) Maximum Inter-arrival Time

100% of inter arrival times are ≤ 1 second

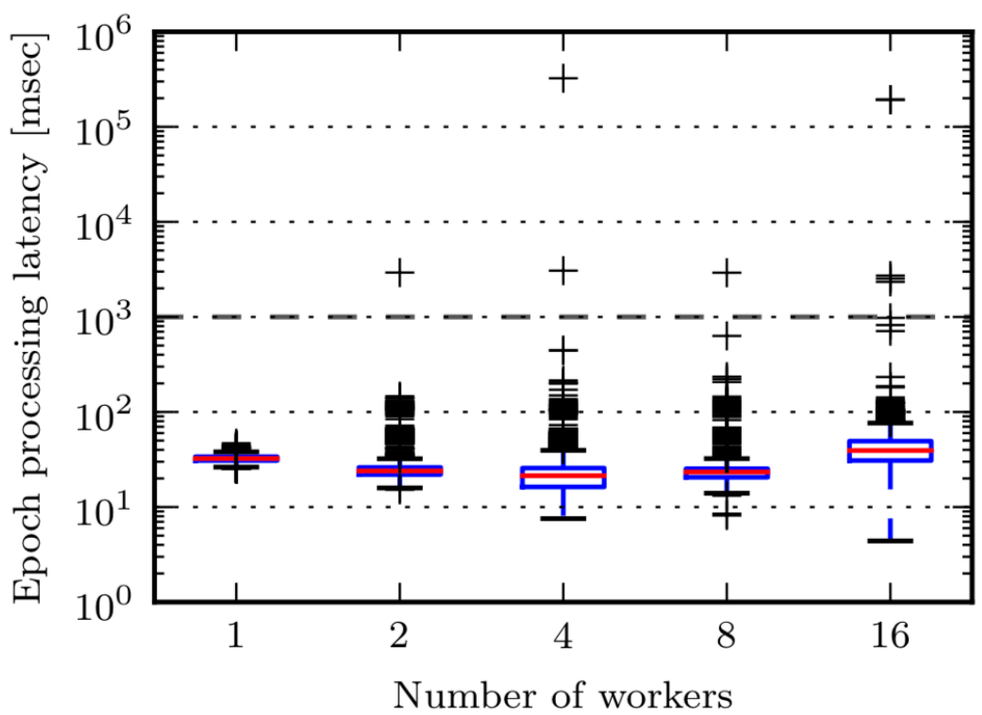
Figure 1. CDFs (Cumulative Distribution Functions) showing total duration of transaction trees and maximum interval between messages of a single session. Note: the x-axis is in logarithmic scale.

REAL-TIME LOG PROCESSING

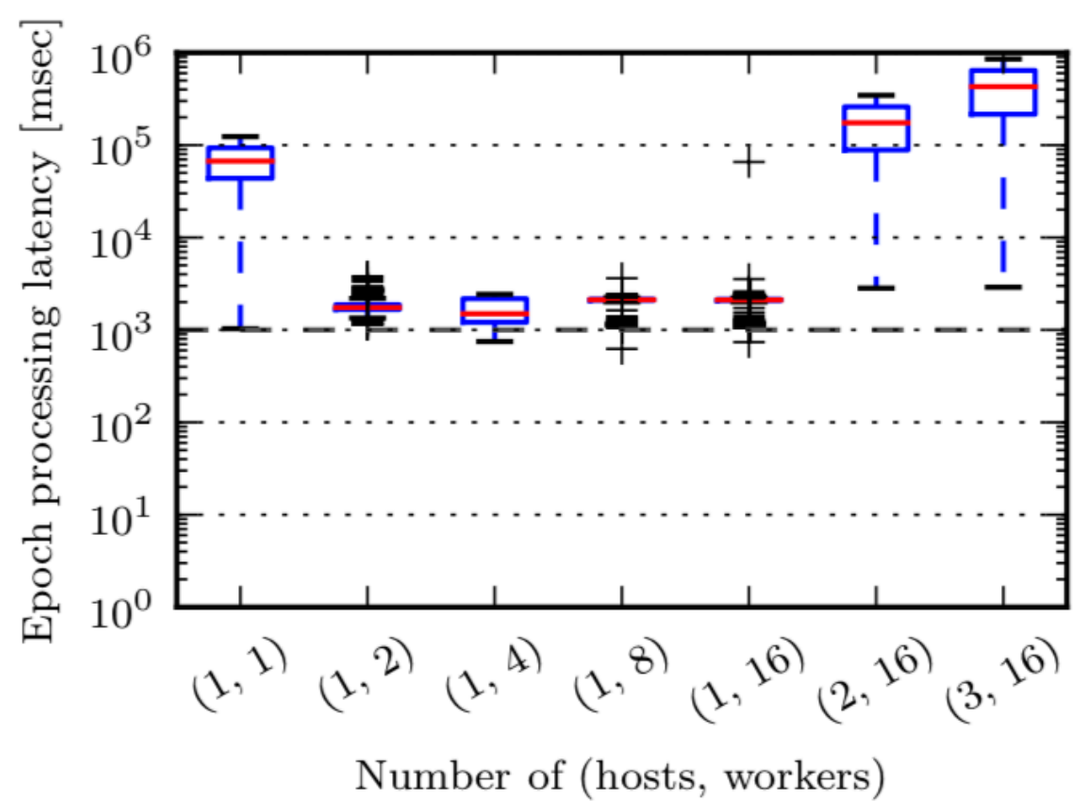
Low latency: Flink spent on average 2.1 seconds (±1.1 s) for processing a single epoch of streaming logs whereas our system took only 26 milliseconds (±53 ms)

Peak resident set size remained stable and reached a peak of 203 MB while Flink's heap rose above 7.5 GB and required considerable tuning

Strymon

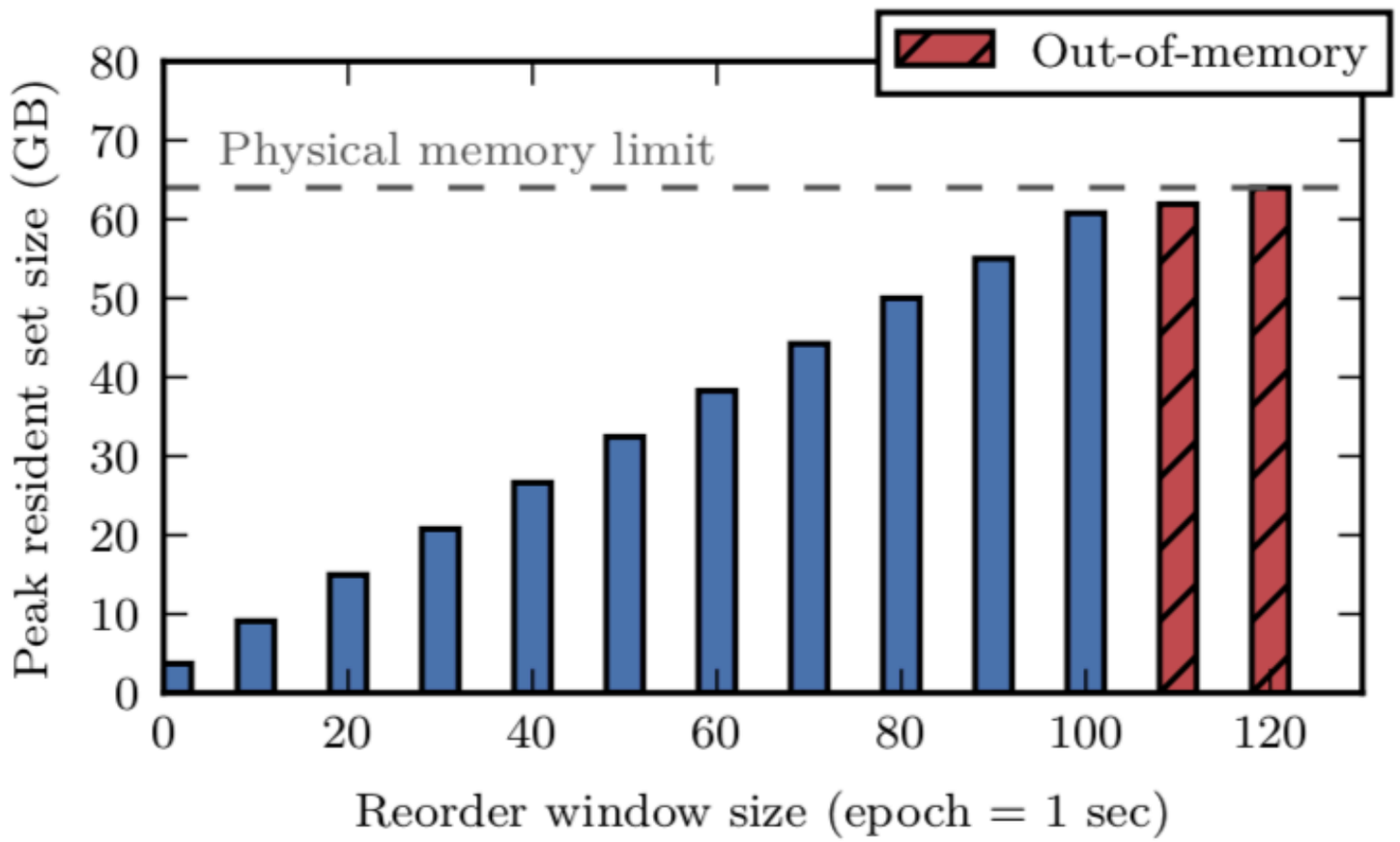


Apache Flink

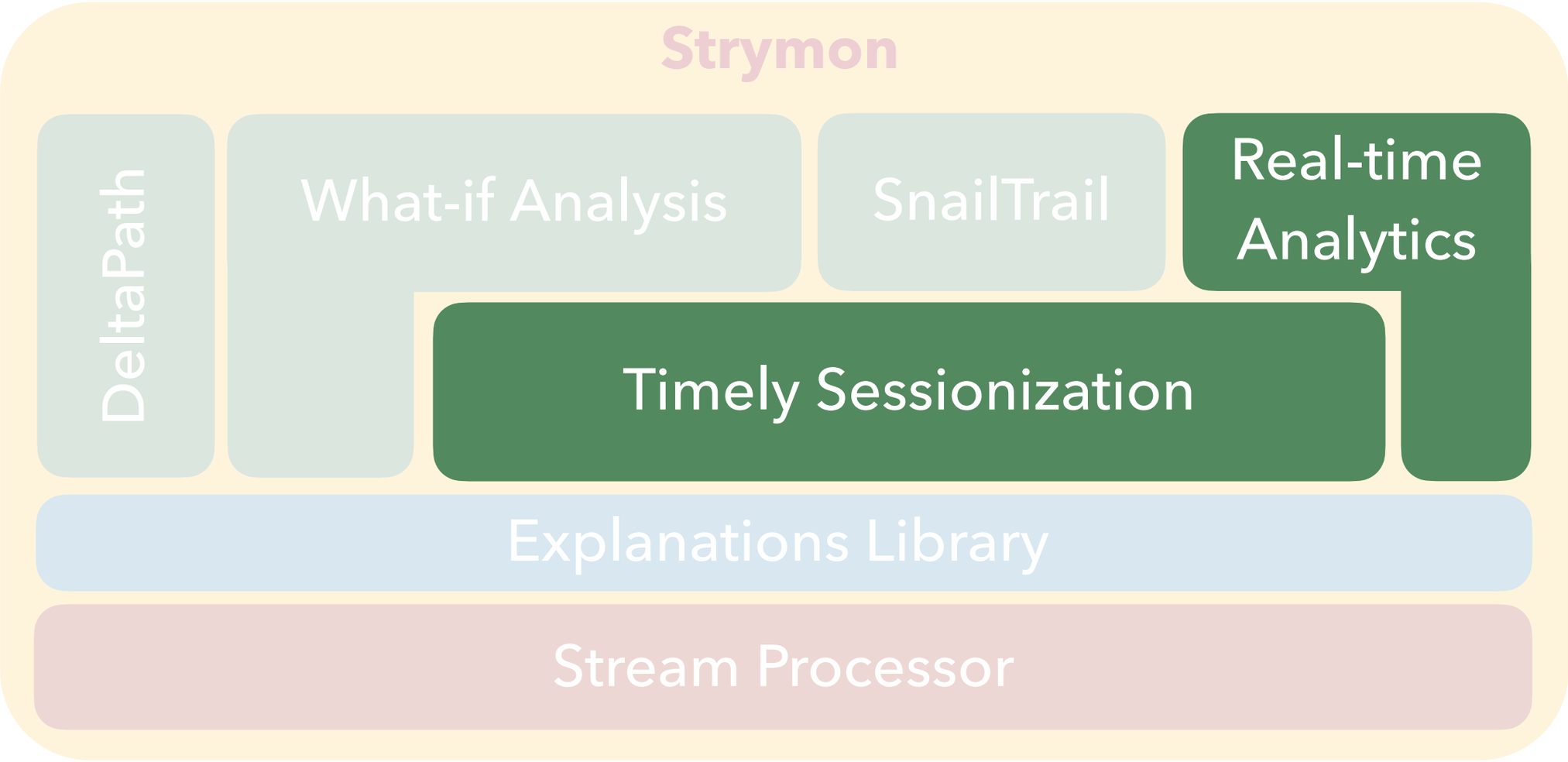


STRYMON'S MEMORY CONSUMPTION

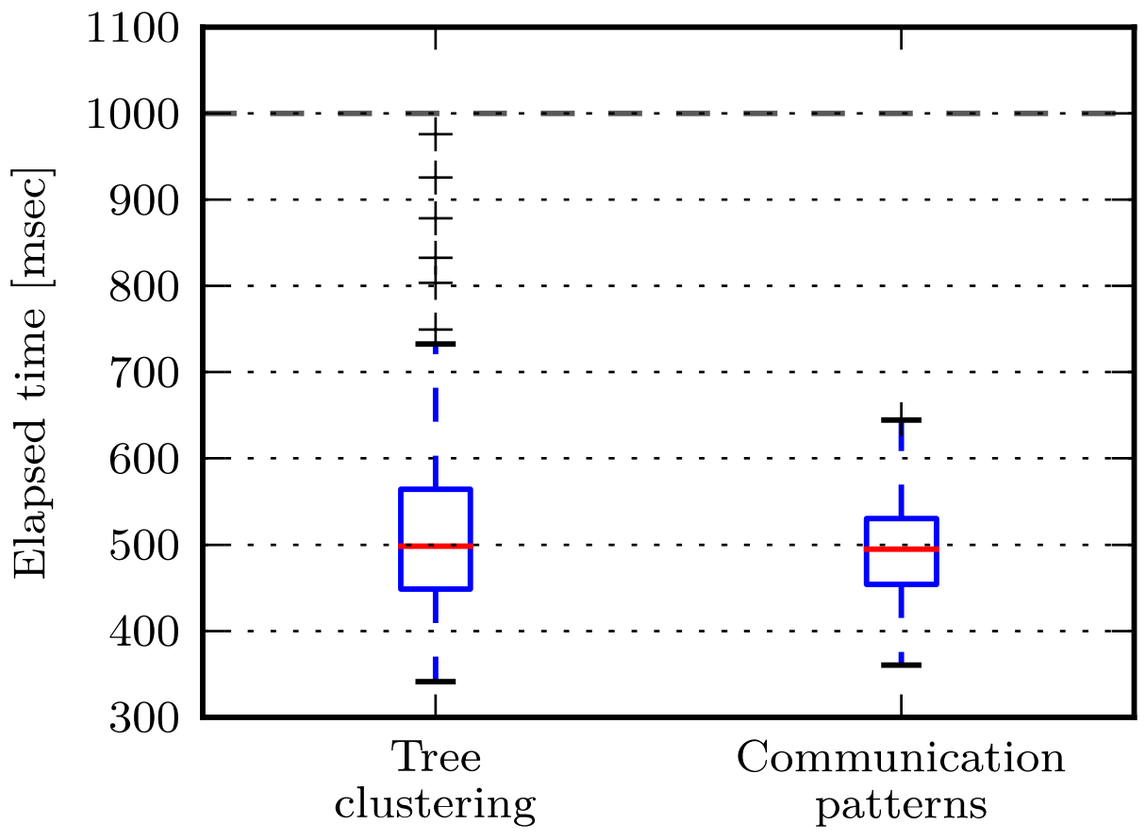
Linear memory increase wrt the size of the reorder buffer



COMPOSITION OF ANALYTICAL WORKFLOWS



ANALYTICS ON THE OUTPUT OF SESSIONIZATION



Exploiting a general framework permits a simple, concise implementation in 1770 lines of code while seamlessly integrating with management applications.

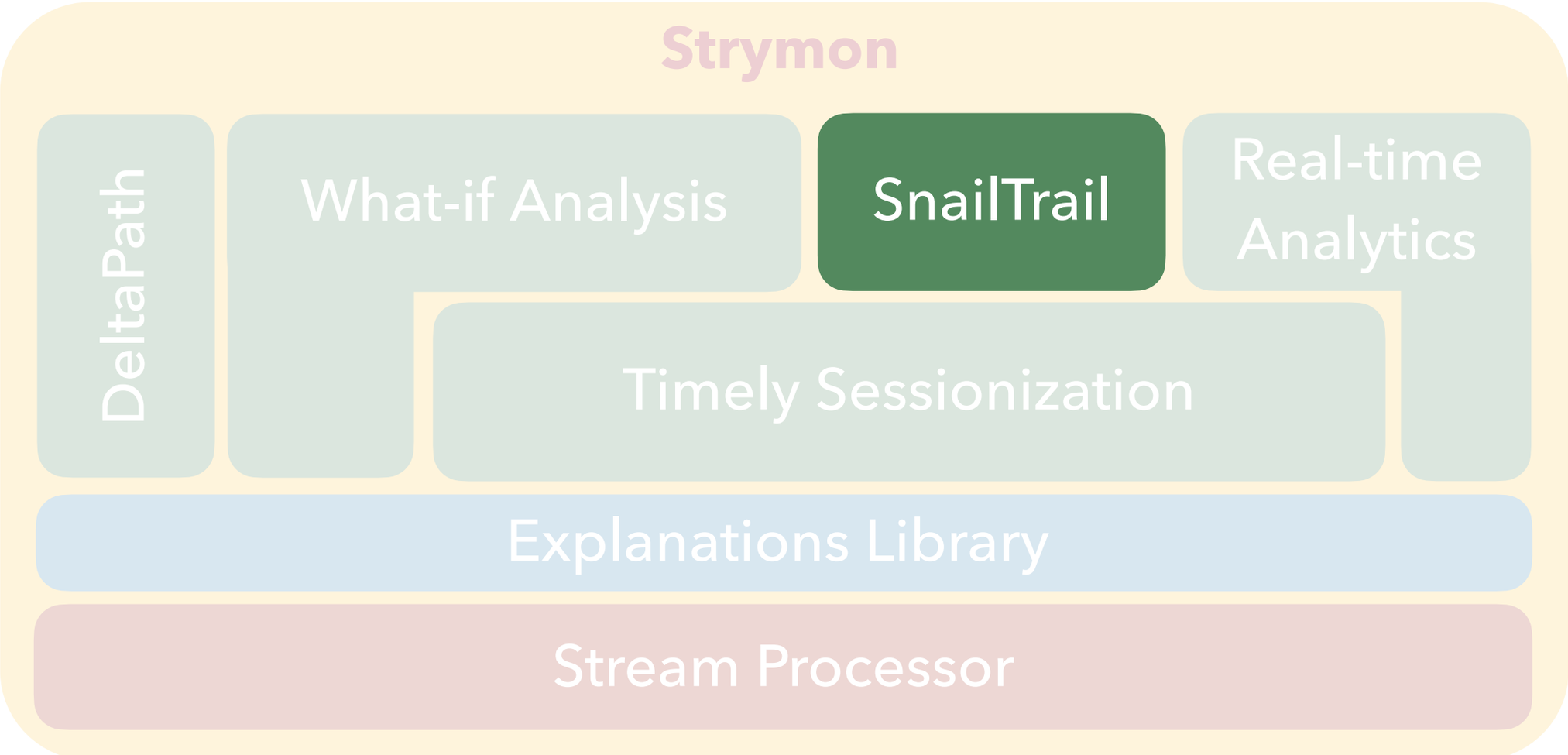
Composition of analytic tasks:

- Online trace tree clustering
- Service dependency extraction
- Inferring call-graph patterns

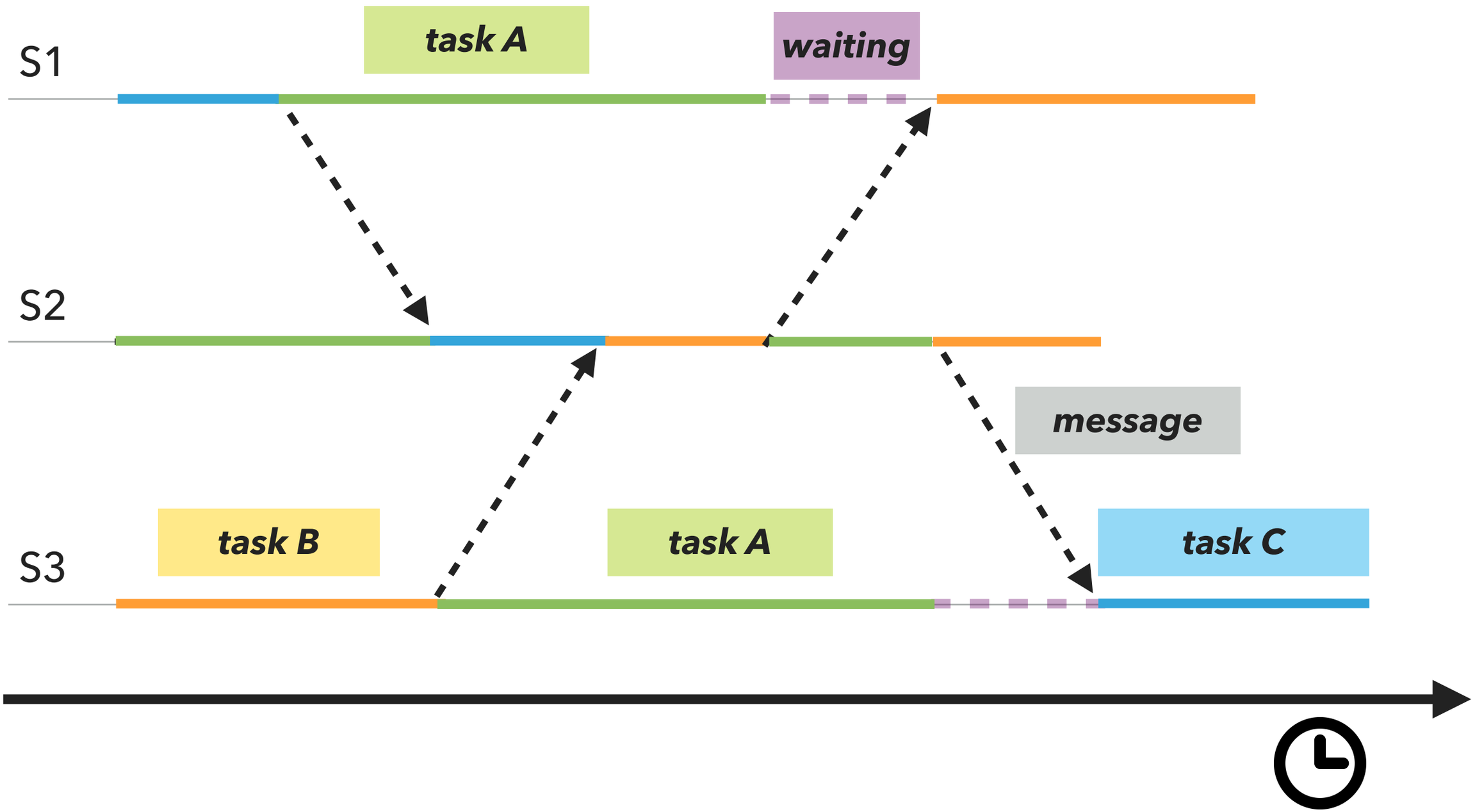
COMPOSING CONTINUOUS STREAMING DATAFLOWS

```
let (input, stream) = scope.new_input();

// Construct session summaries
let trees = stream.sessionize(INACTIVITY_LIMIT)
                  .construct_trace_trees();

// Trace tree durations
trees.filter(|t| t.messages.len() >= 2)
     .map(|t| min_max_time(t.messages))
     .histogram(|x| log_discretize(x));

// Classify trace trees by structure
trees.map(|t| t.signature())
     .topk(|x| x.clone(), 10);


// Identify pairs of communicating services
trees.flat_map(|t| service_call_patterns(t)
     .topk(|pairs| pairs.clone(), 10)
     .show_each_epoch();
```

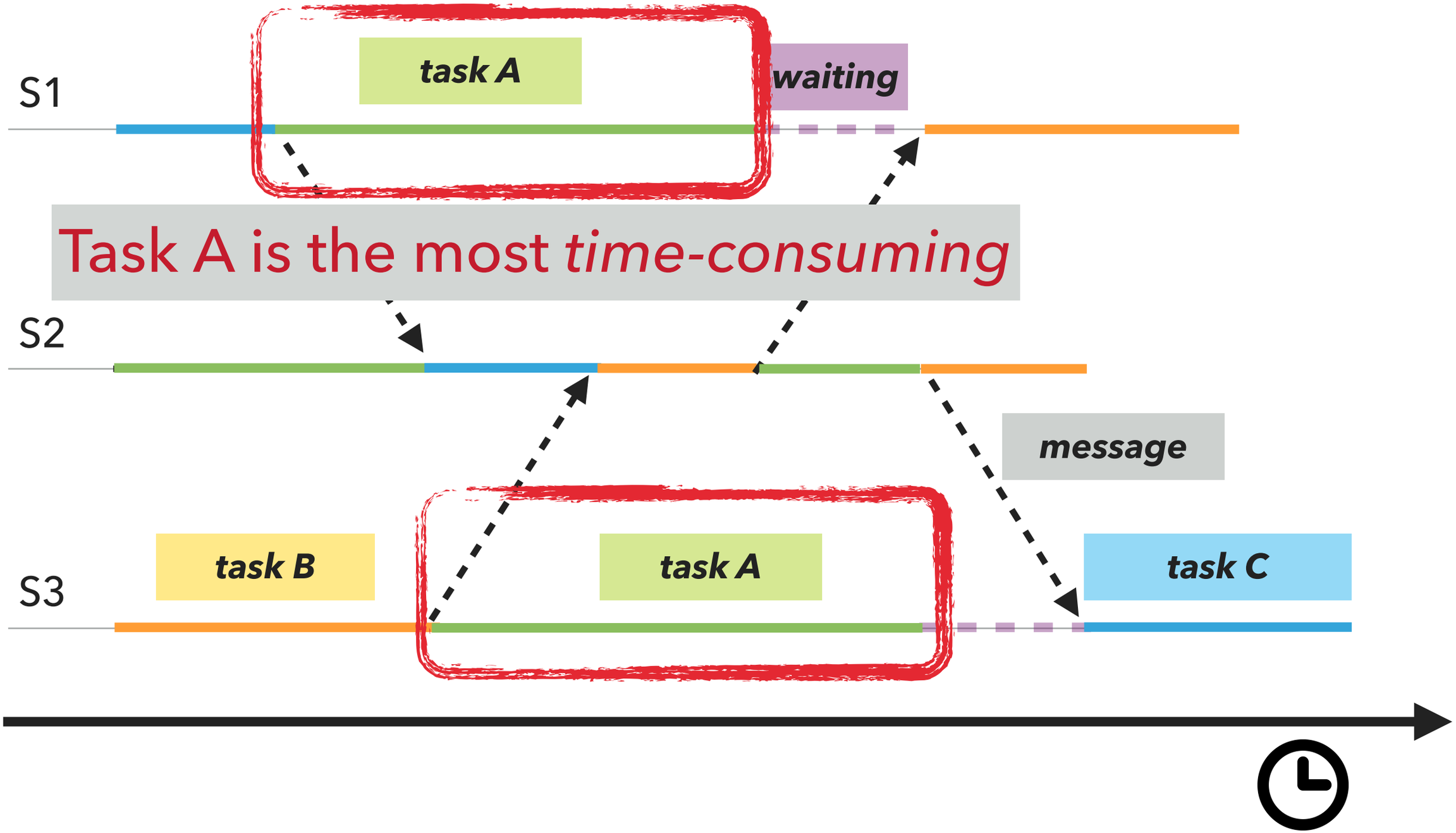
ONLINE CRITICAL PATH ANALYSIS



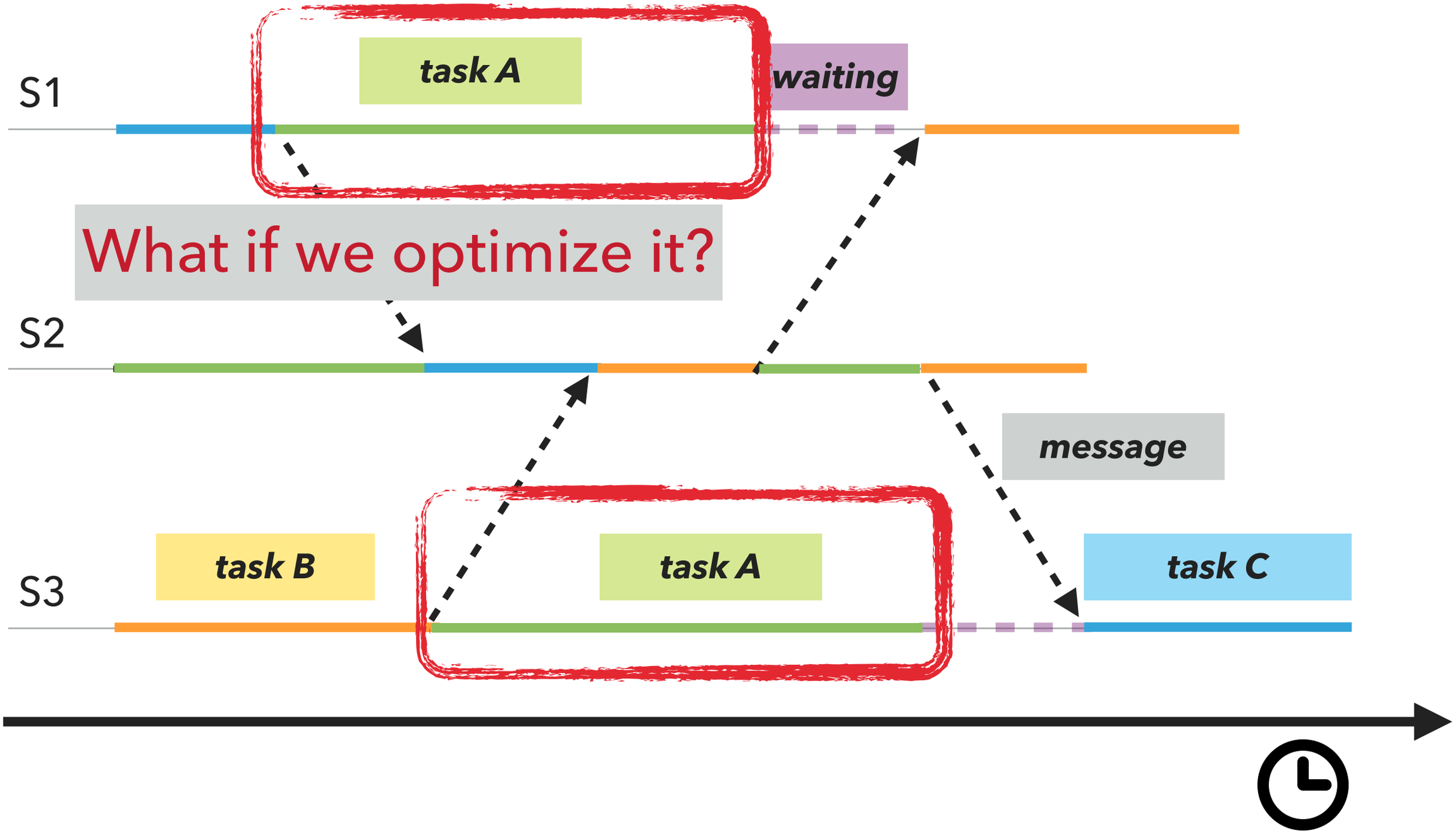
A PARALLEL EXECUTION



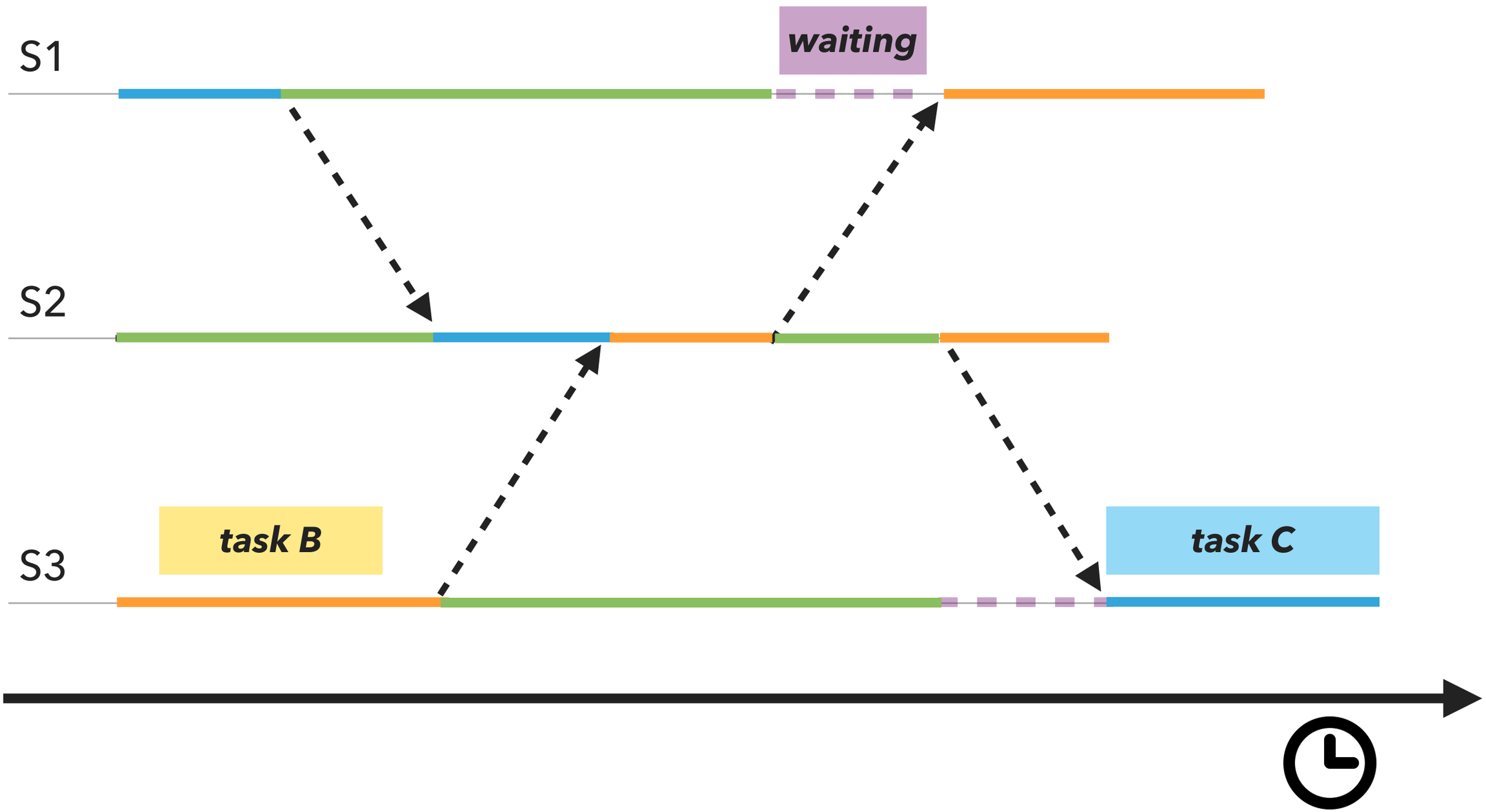
A PARALLEL EXECUTION



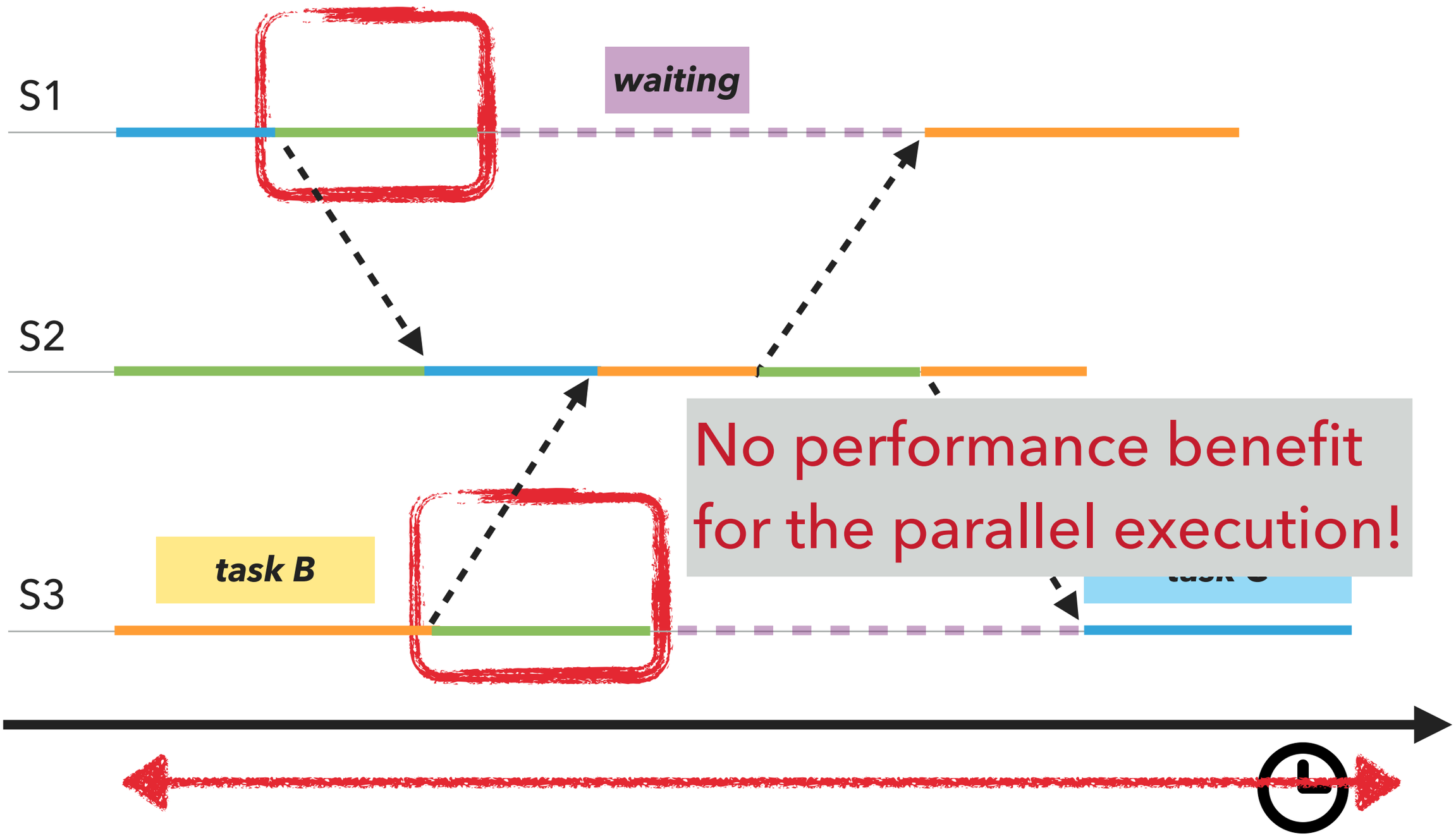
A PARALLEL EXECUTION



A PARALLEL EXECUTION



A PARALLEL EXECUTION

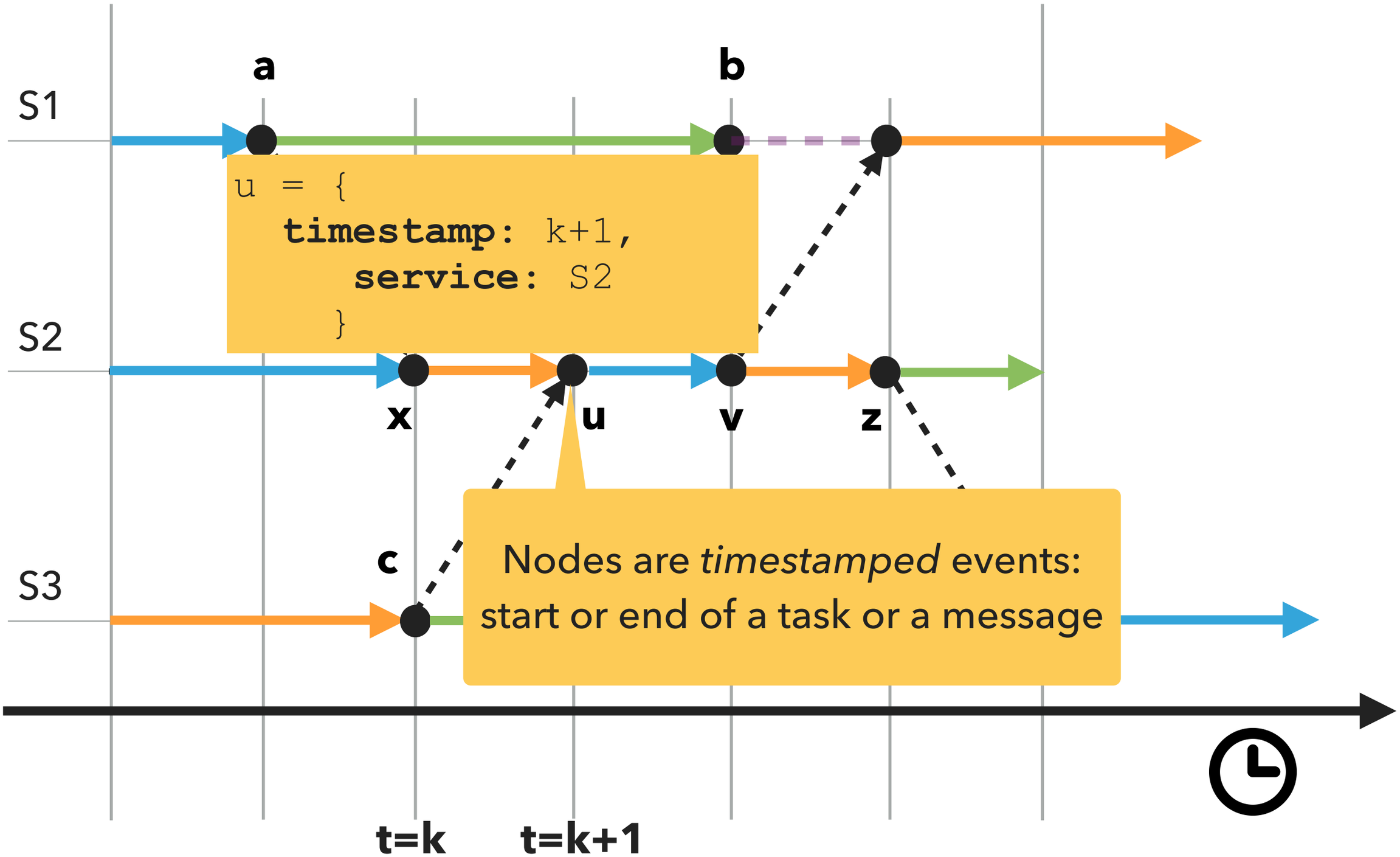


No performance benefit for the parallel execution!

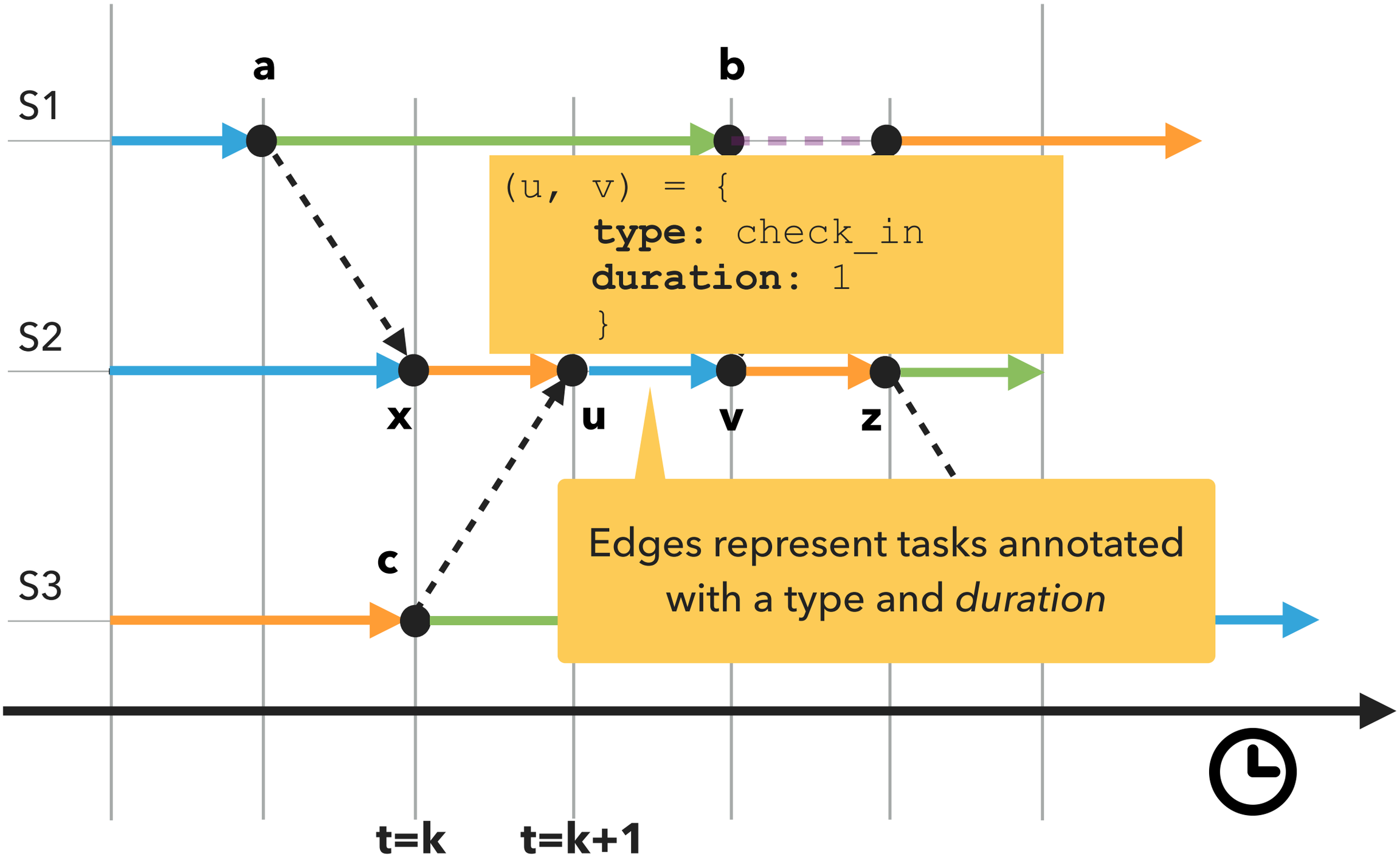
A PARALLEL EXECUTION

conventional profiling can be misleading

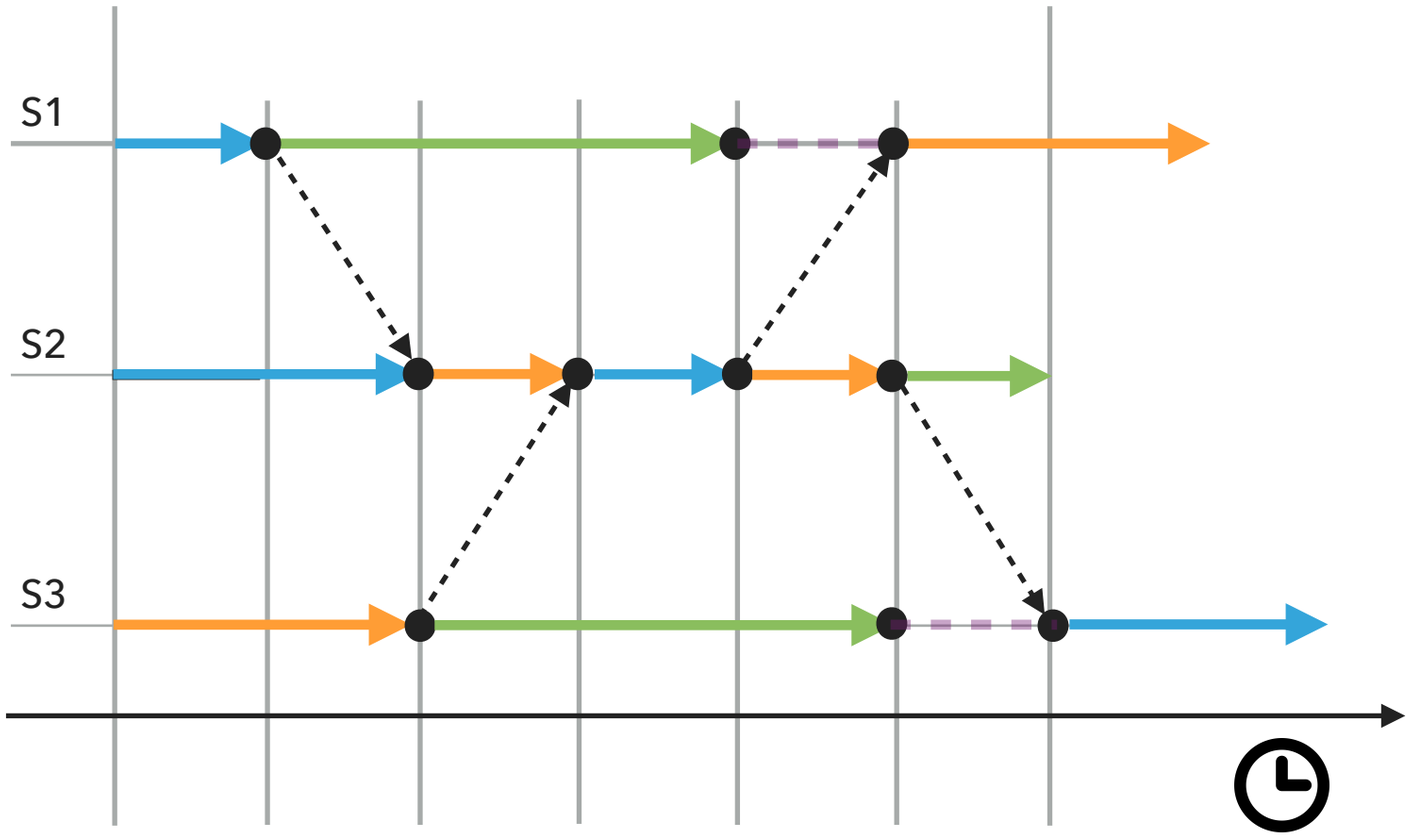
THE PROGRAM ACTIVITY GRAPH (PAG)



THE PROGRAM ACTIVITY GRAPH (PAG)

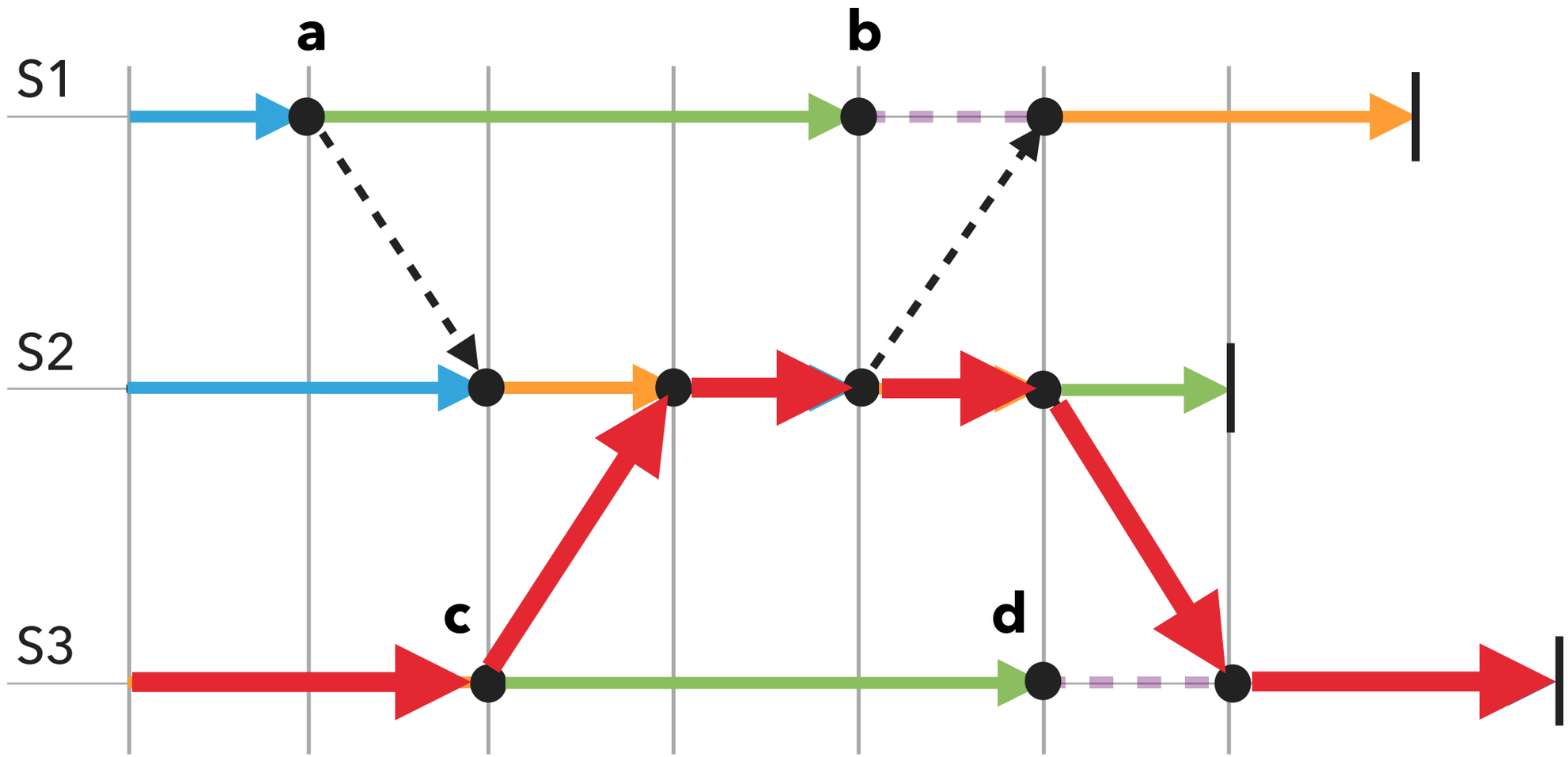


The Program Activity Graph captures computational dependencies



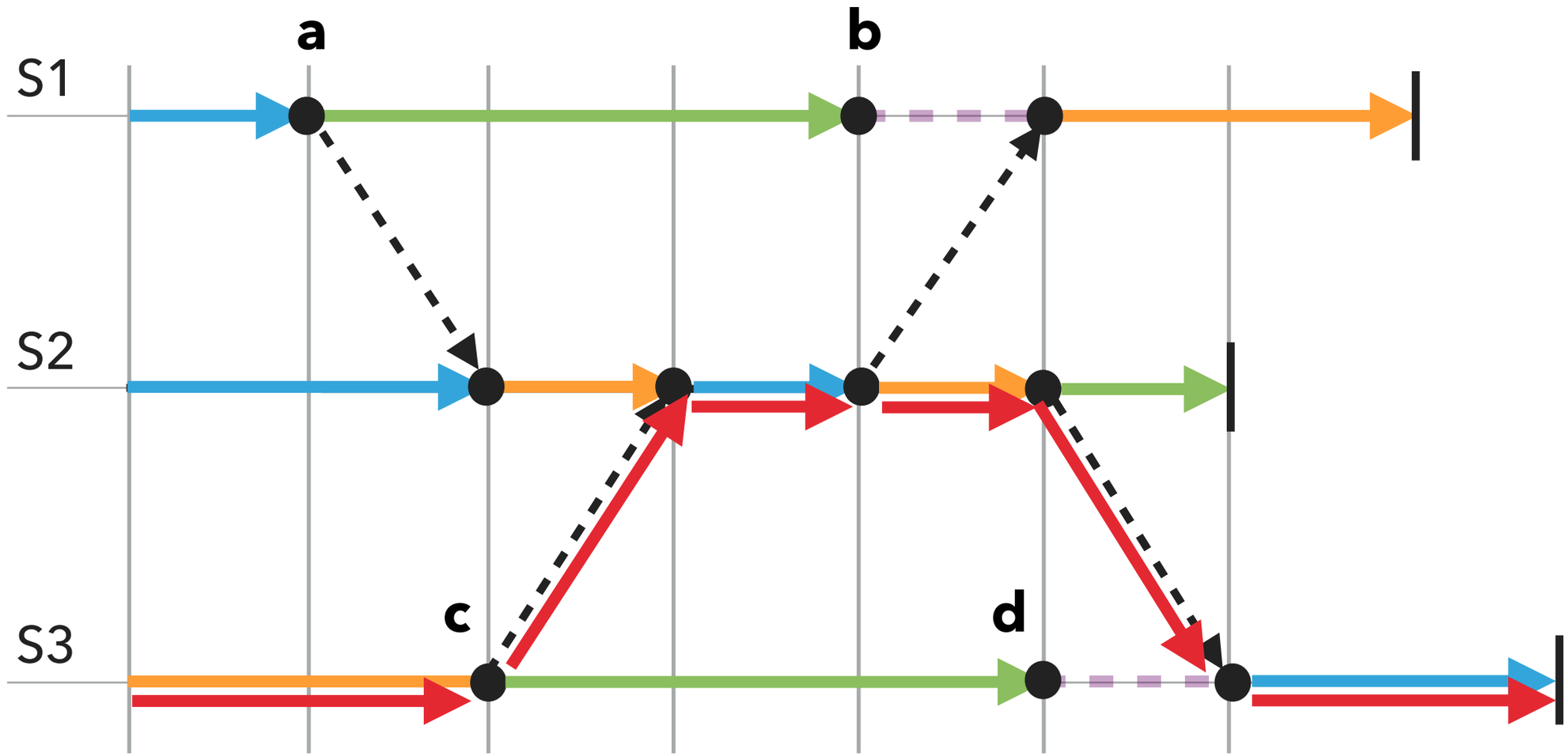
- Which tasks delay the overall execution?
- i.e. which tasks lie on the critical path of the execution?

CRITICAL PATH



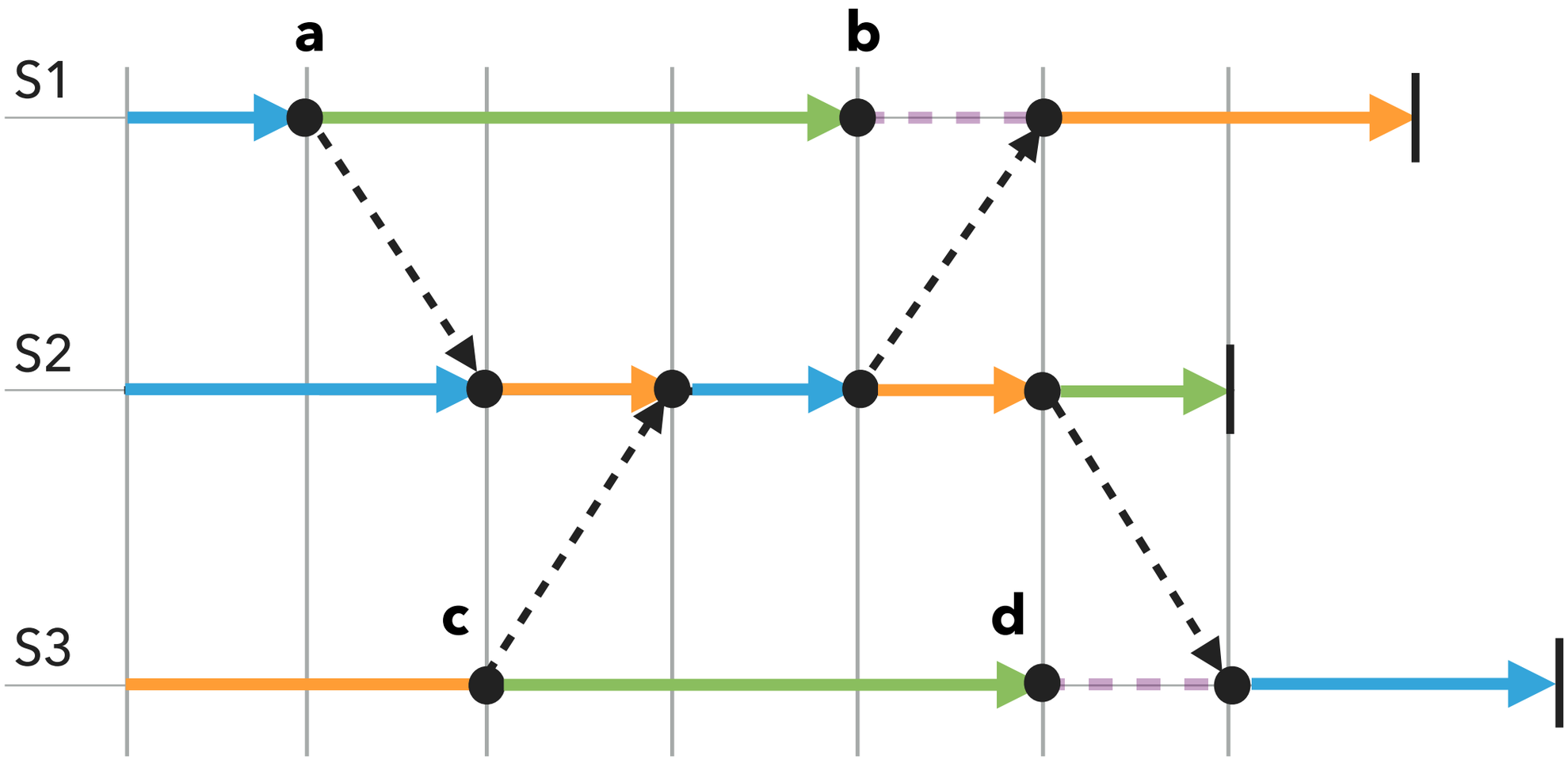
The longest path in the execution history
(not considering waiting activities)

CRITICAL PATH

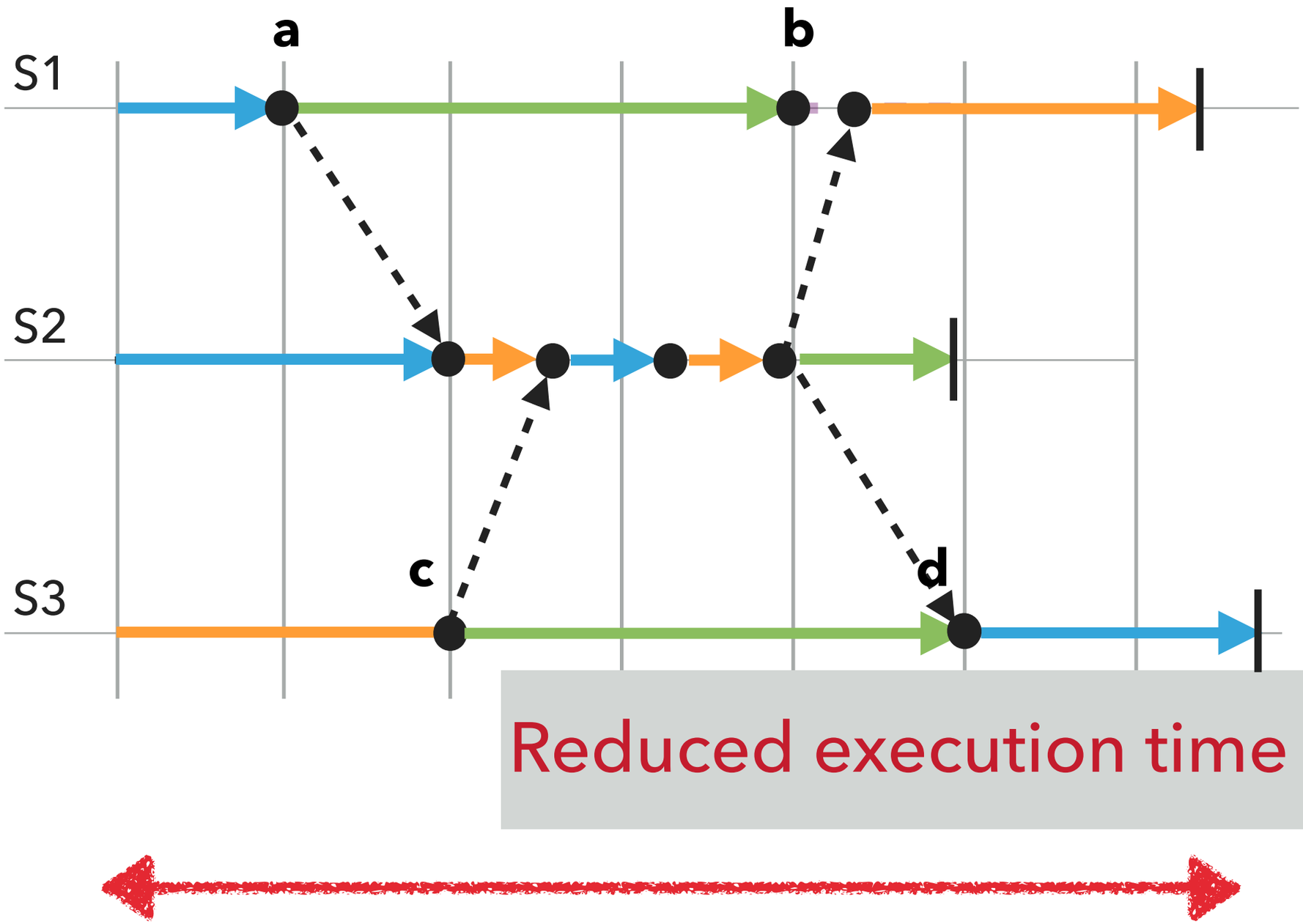


Optimizing activities on the critical path can potentially reduce the execution time

CRITICAL PATH

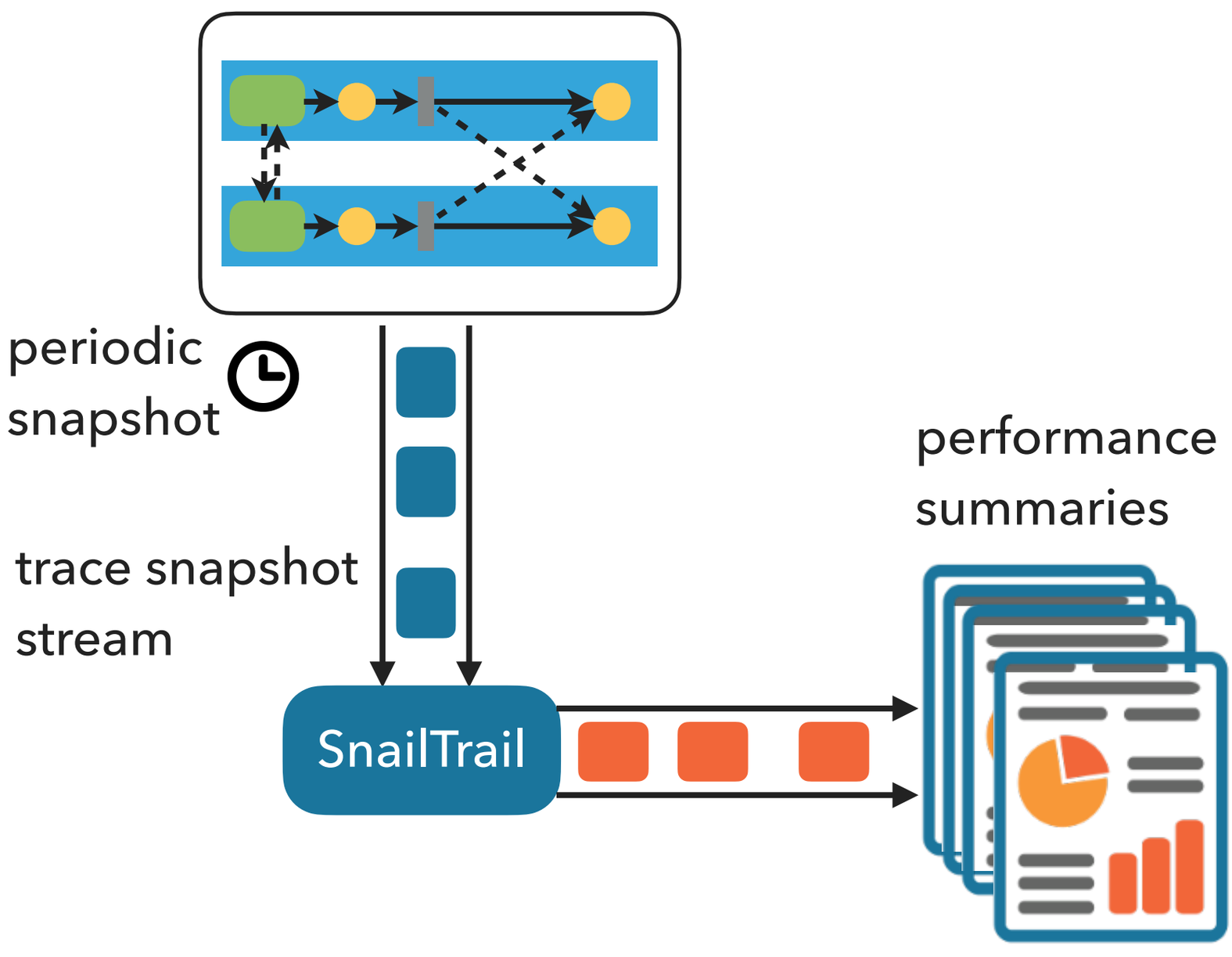


CRITICAL PATH



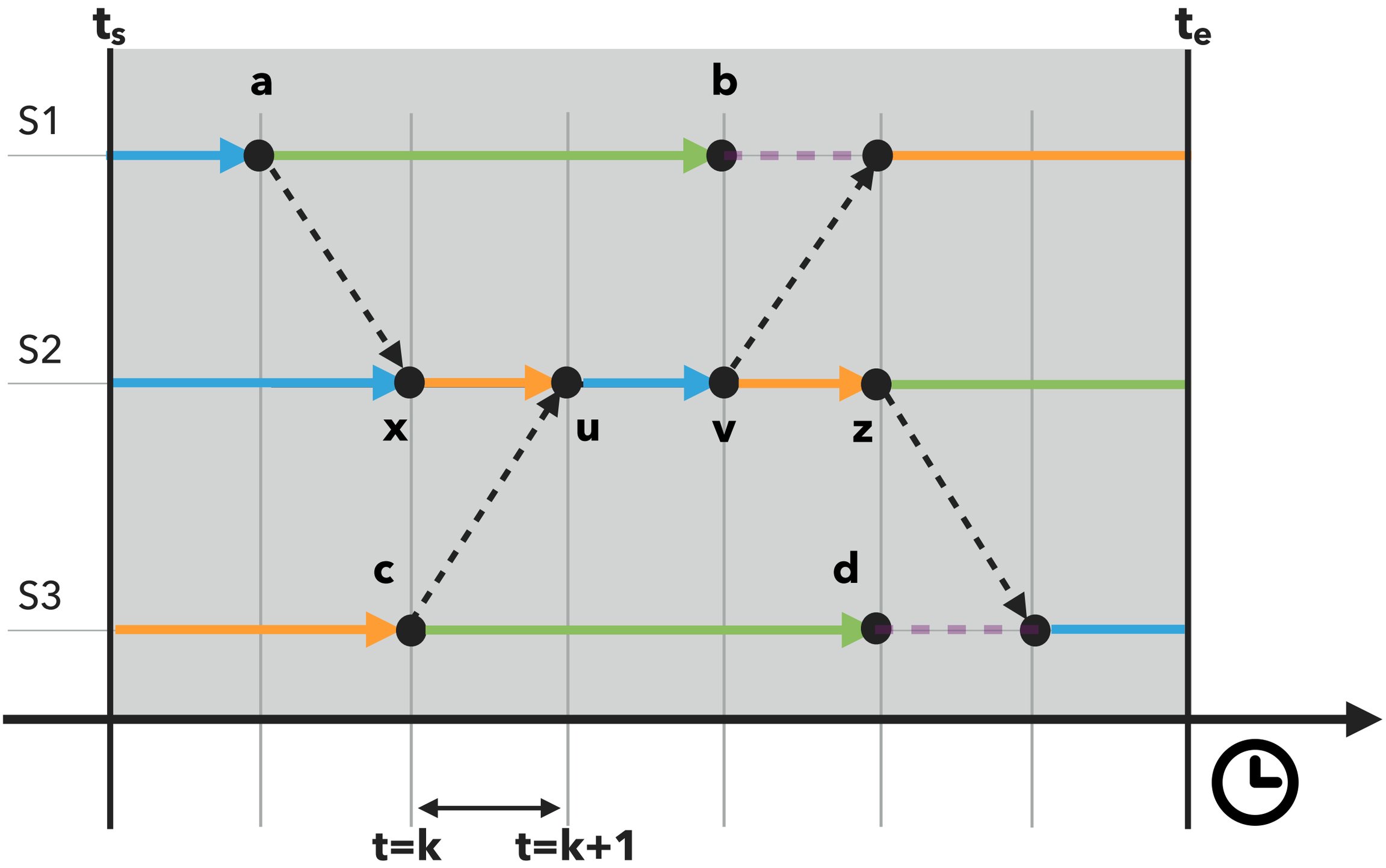
How to compute the critical path of distributed applications in an online fashion?

ONLINE ANALYSIS OF TRACE SNAPSHOTS

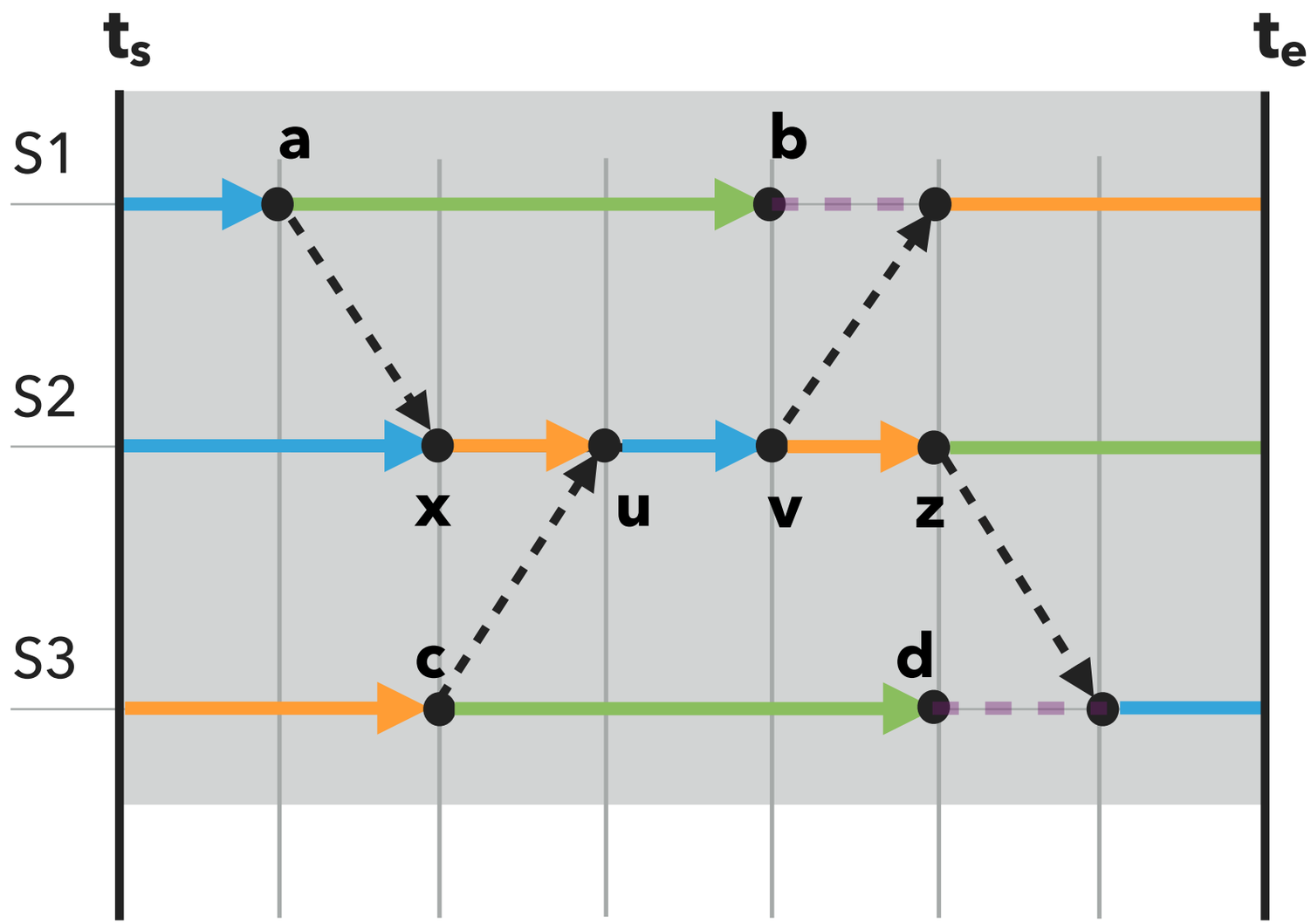


https://github.com/strymon-system/snailtrail

PROGRAM ACTIVITY GRAPH SNAPSHOT



All paths are potentially part of the evolving critical path

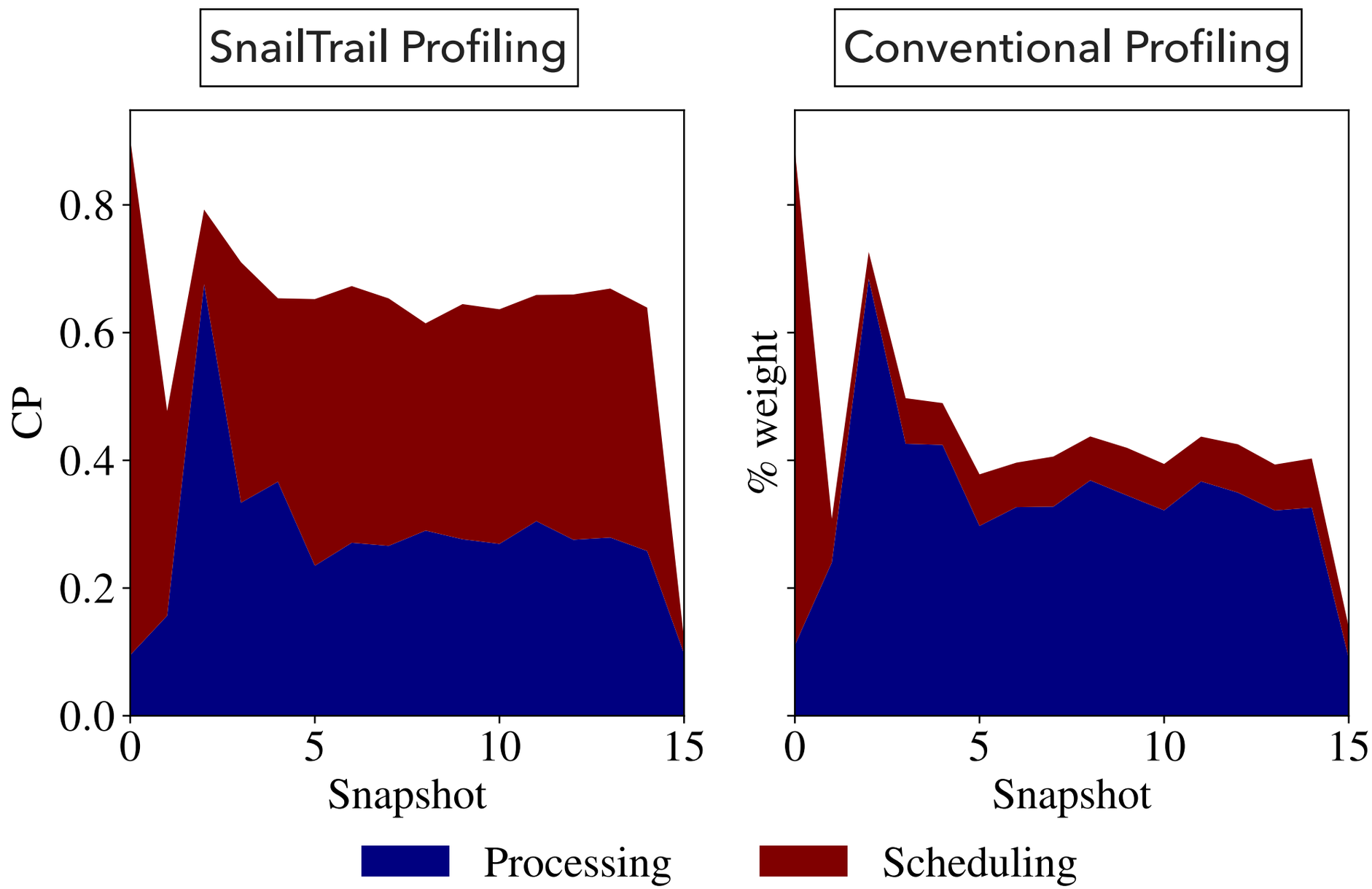


How to rank tasks with regard to criticality?

Intuition: the more paths a task appears on the more probable it is critical

USE CASE: DISTRIBUTED DATAFLOWS

COMPARISON WITH CONVENTIONAL PROFILING



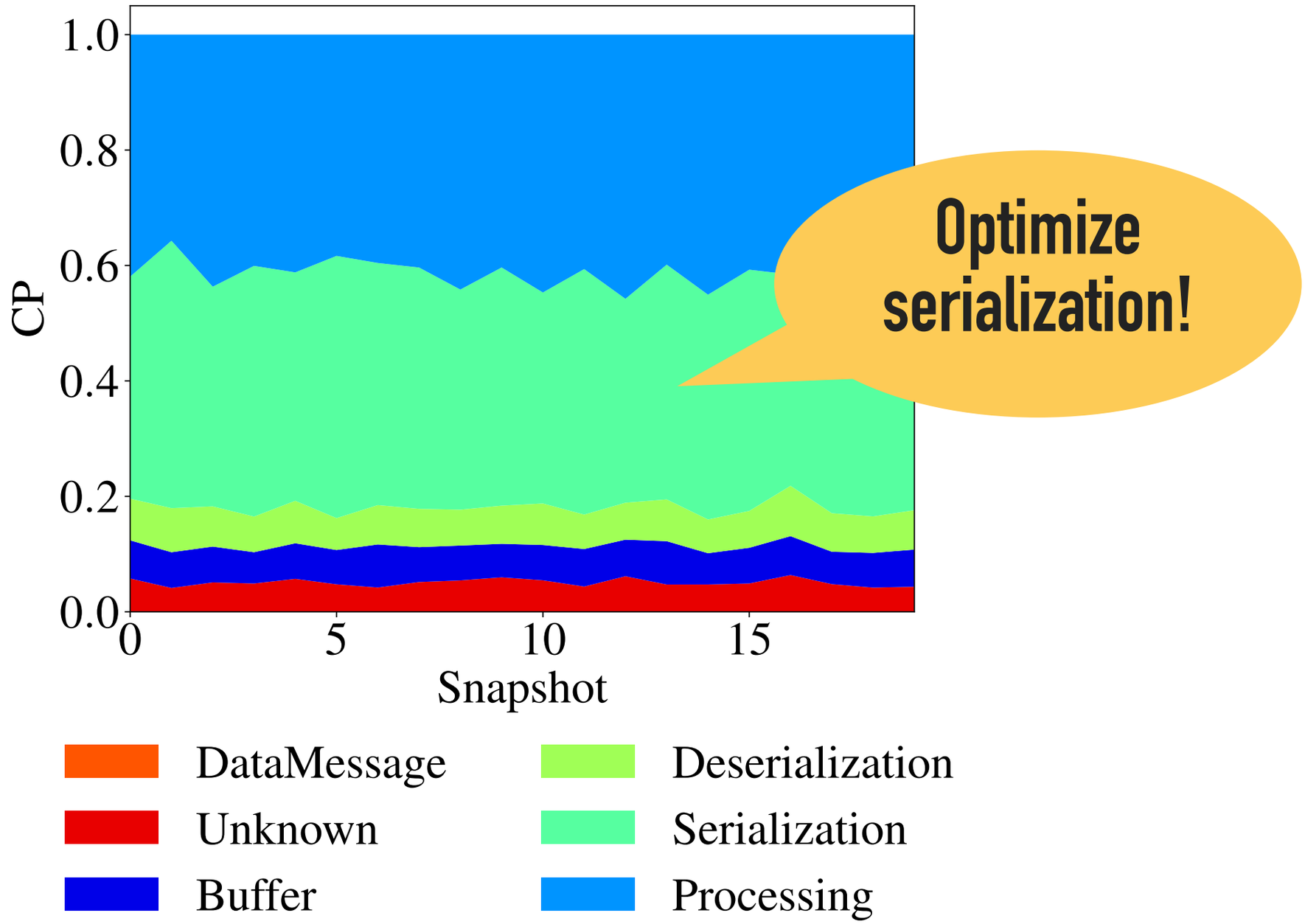
Apache Spark: Yahoo! Streaming Benchmark, 16 workers, 8s snapshots

See also: Venkataraman, S., Panda, A., Ousterhout, K., Ghodsi, A., Franklin, M. J., Recht, B., and Stoica, I. *Drizzle: Fast and Adaptable Stream Processing at Scale*. In SOSP, 2017.

SNAILTRAIL SUMMARIES

- Activity Summary
  - *which **activity type** is a bottleneck?*

Activity Summary



Apache Flink: Dhalion WordCount Benchmark, 4 workers, 1s snapshots

SNAILTRAIL SUMMARIES

- Activity Summary
  - *which **activity type** is a bottleneck?*
- Straggler Summary
  - *which **worker** is a bottleneck?*

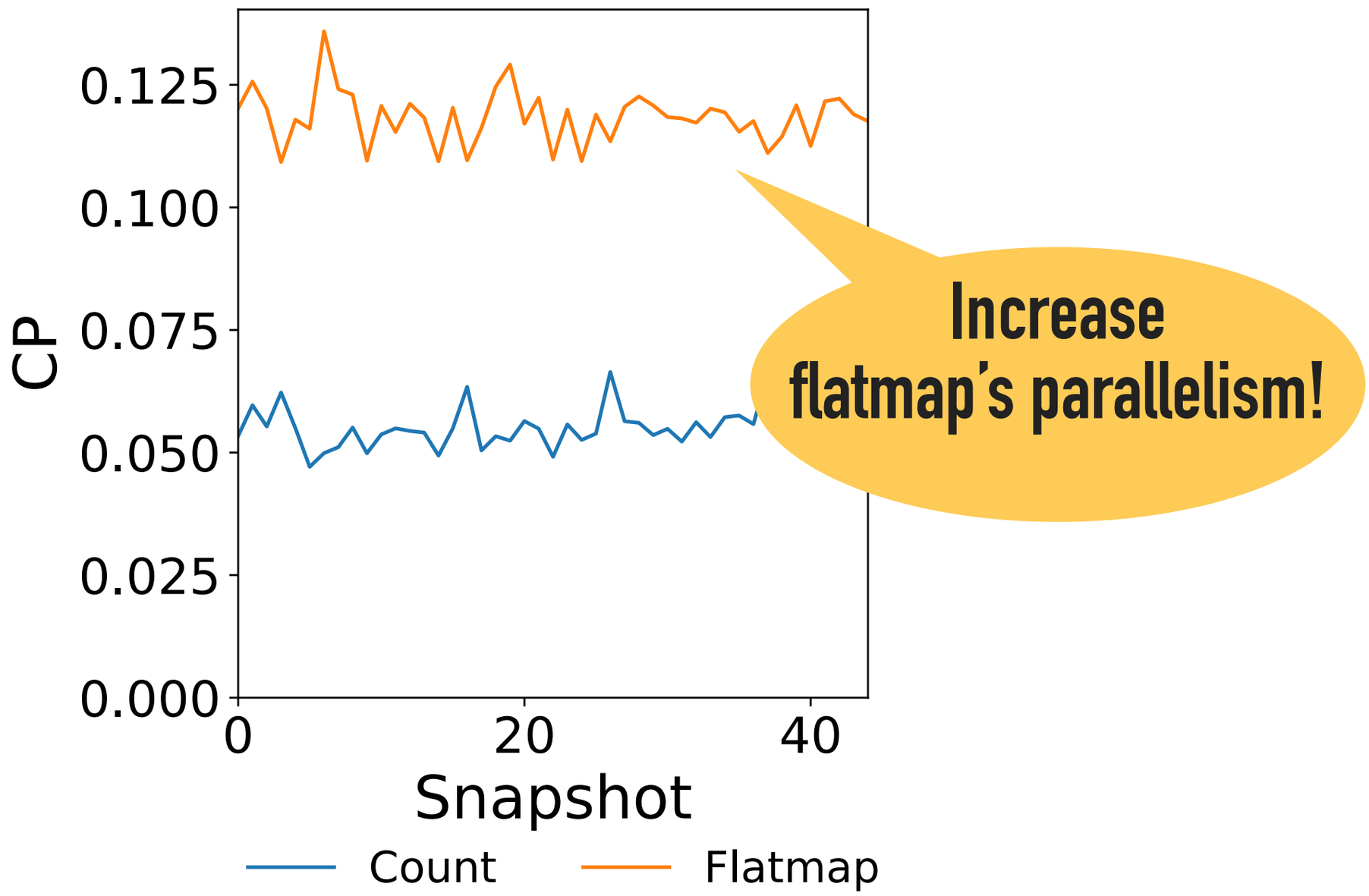
Straggler Summary

Apache Flink: Dhalion WordCount Benchmark, 4 workers, 1s snapshots

SNAILTRAIL SUMMARIES

- Activity Summary
  - *which **activity type** is a bottleneck?*
- Straggler Summary
  - *which **worker** is a bottleneck?*
- Operator Summary
  - *which **operator** is a bottleneck?*

Operator Summary

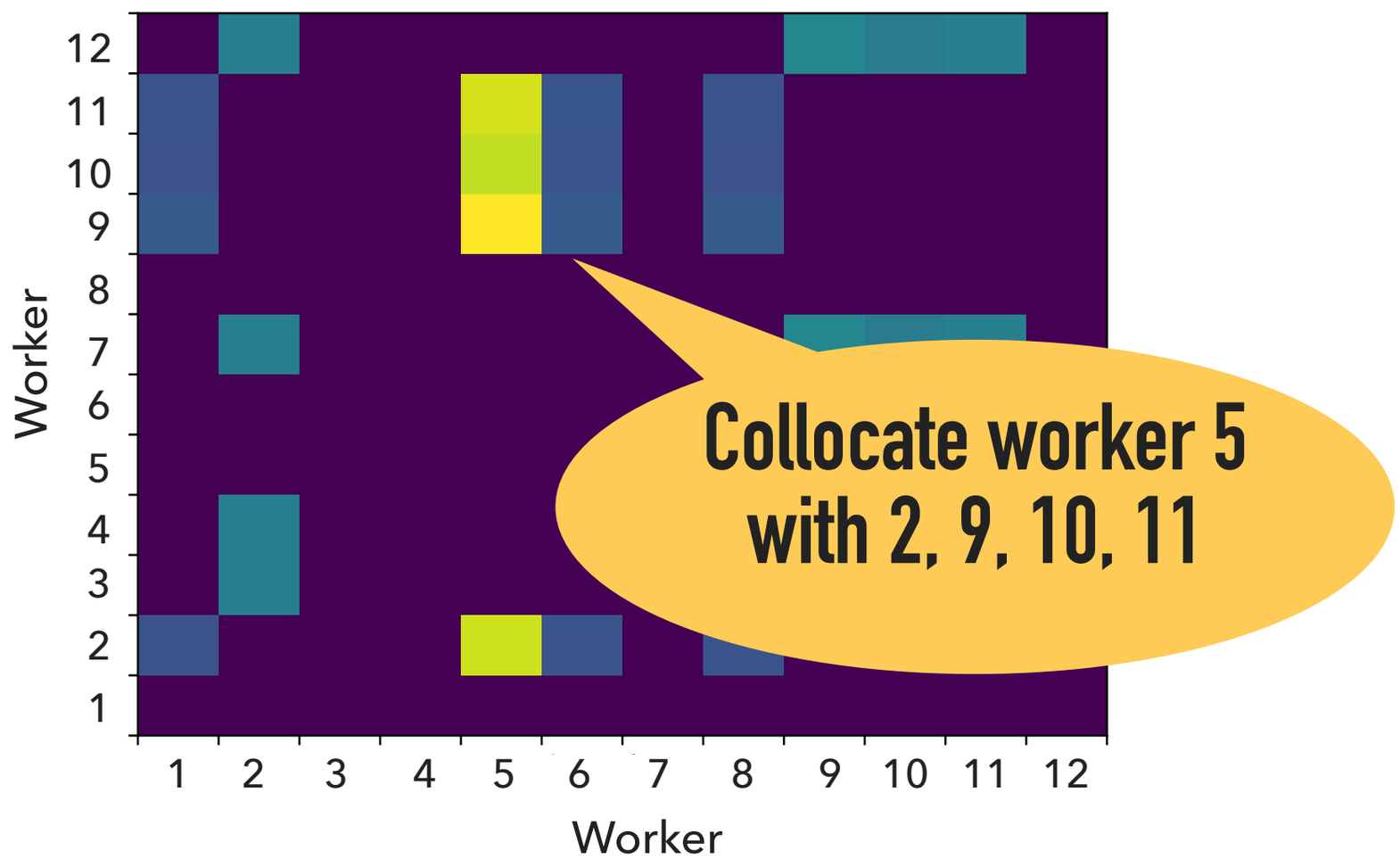


Apache Flink: Dhalion WordCount Benchmark, 10 workers, 1s snapshots

SNAILTRAIL SUMMARIES

- Activity Summary
  - *which **activity type** is a bottleneck?*
- Straggler Summary
  - *which **worker** is a bottleneck?*
- Operator Summary
  - *which **operator** is a bottleneck?*
- Communication Summary
  - *which communication **channels** are bottlenecks?*

Communication Summary

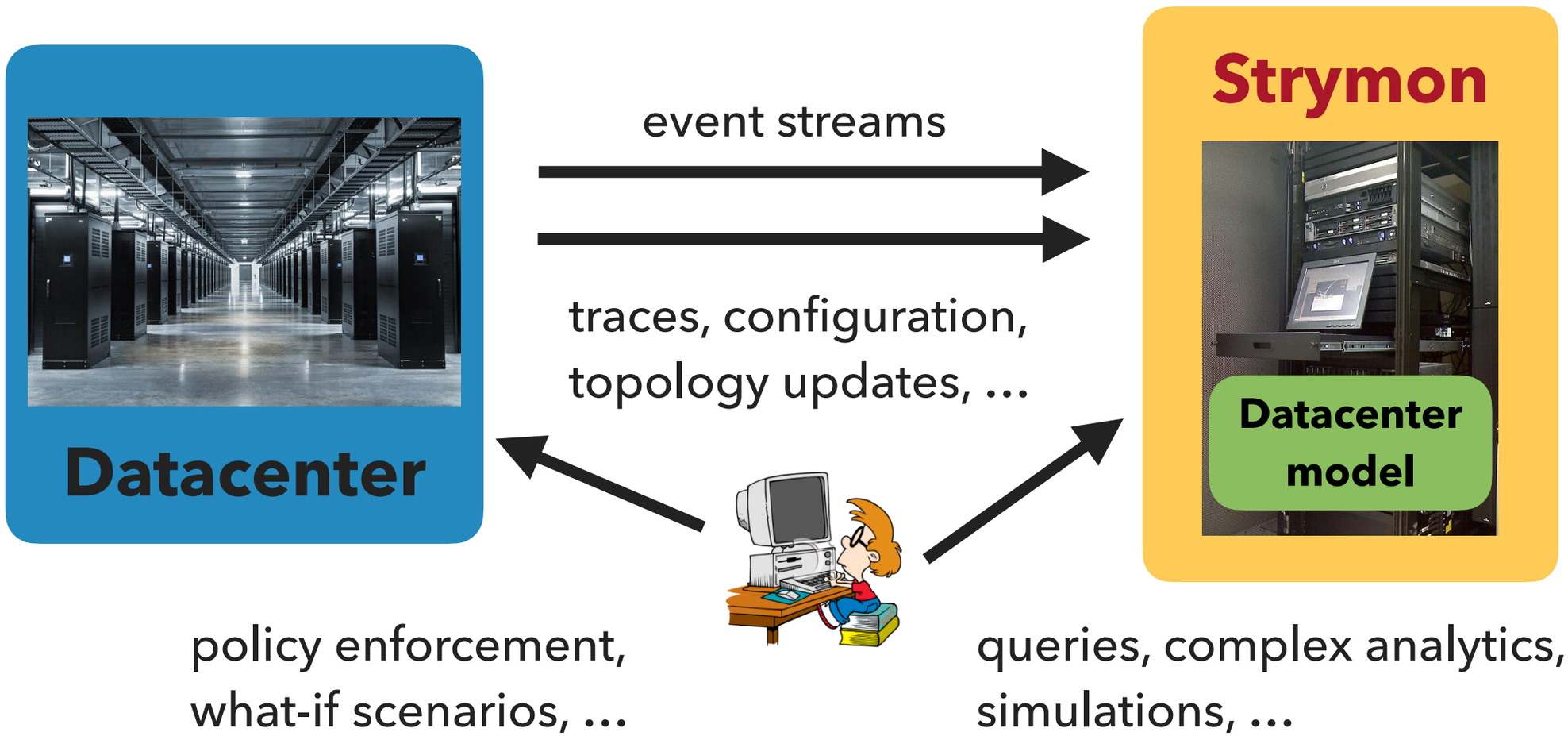


Communication Criticality

Strymon: Breadth-First Search, 12 workers, 1s snapshots

COLLABORATORS

THE STRYMON SYSTEM FOR DATACENTER MANAGEMENT



strymon.systems.ethz.ch