WREN: A Fast and Scalable Transactional Causally Consistent Geo-Replicated Key-Value Store

Diego Didona, Kristina Spirovska, Willy Zwaenepoel

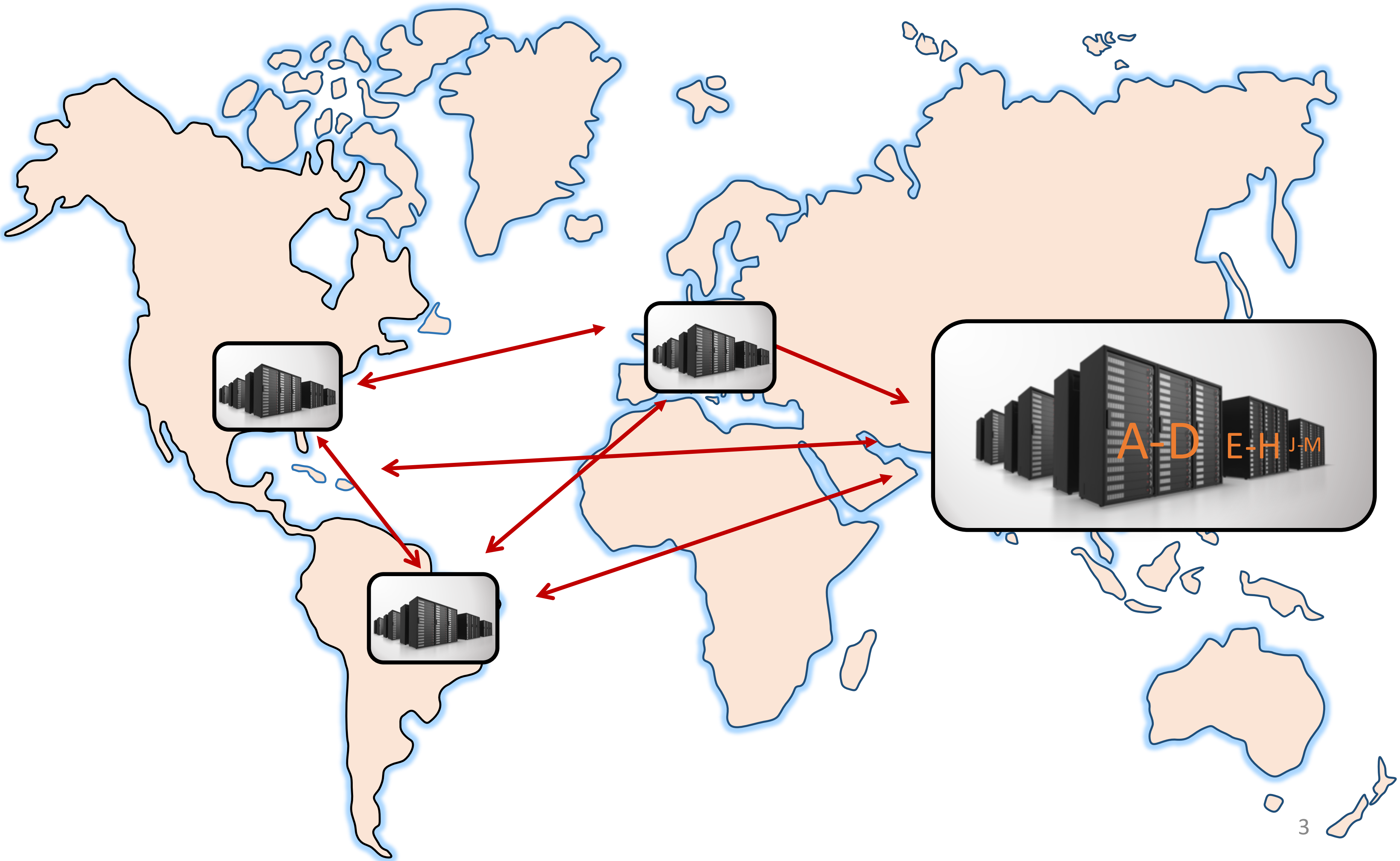
RainbowFS workshop

Paris, May 3rd

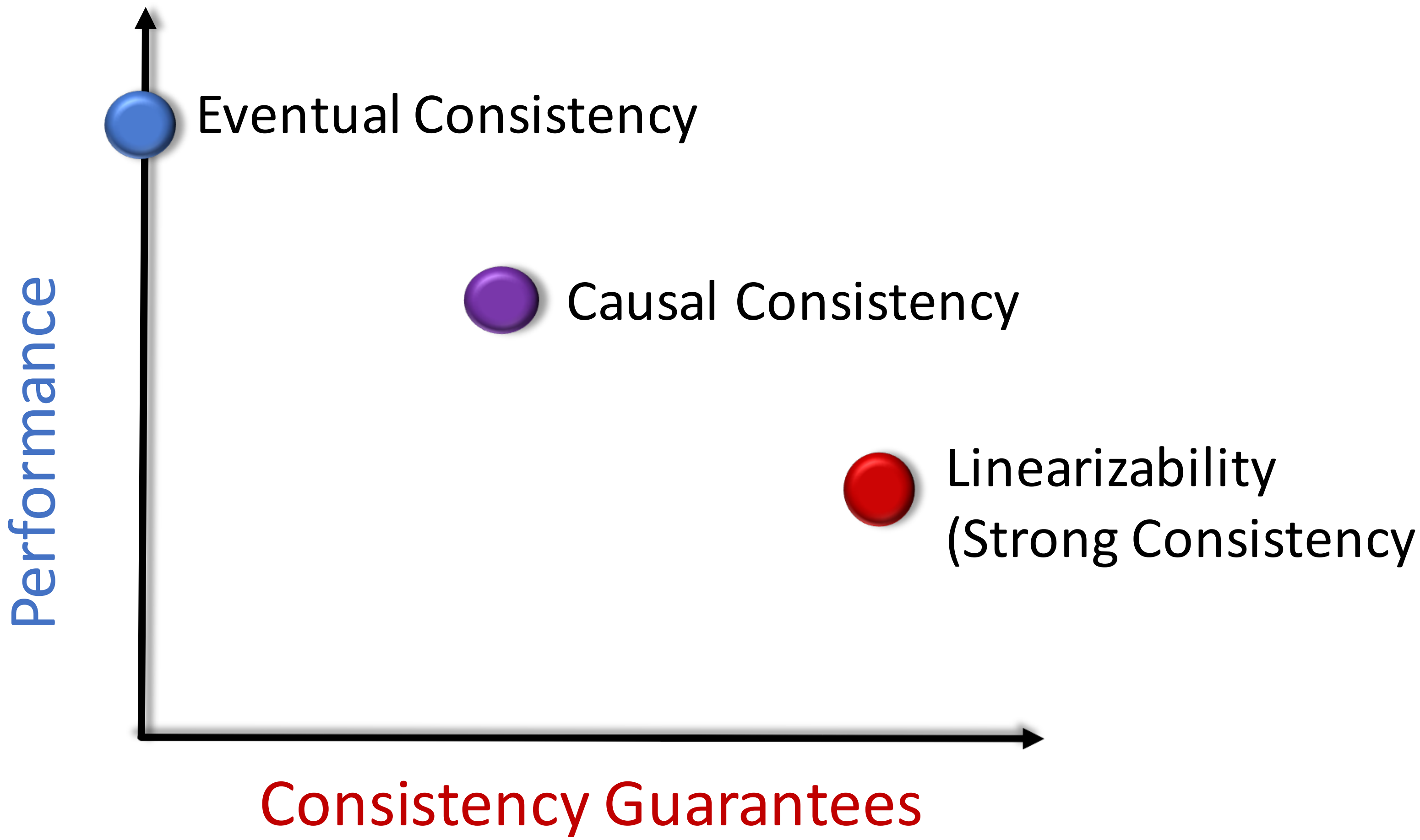
Outline

- Introduction
- Our work- Wren
- Related work
- Summary

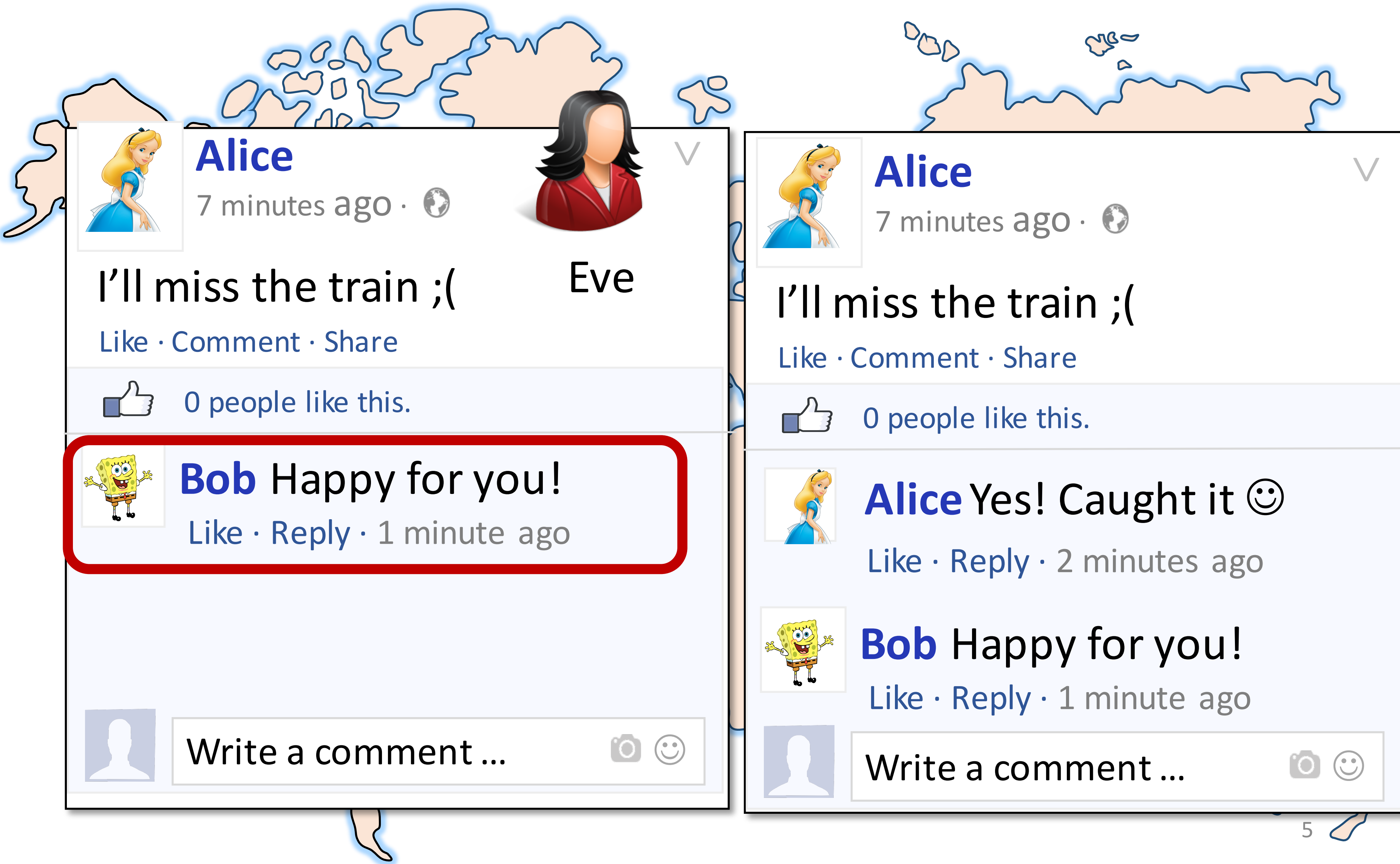
Geo-replicated partitioned key-value data store



Consistency



Causal Consistency



Transactional Causal Consistency

- Transactions read from a **causally consistent snapshot**
- Transactions updates are visible **atomically** in a DC

Outline

- Introduction
- Our work- Wren
- Related work
- Summary

Limitations of State-of-the-art Systems

Dependency tracking

- Metadata scalability

Update timestamping

- Clock skew induced latency

Main Contributions

1. Constant Metadata

Hybrid Stable Time (HST)

A novel dependency tracking and stabilization protocol

Addresses metadata scalability

2. Low Latency

Hybrid Logical/ Physical Clocks

Loose synchronization of physical clocks without suffering from clock skew

Addresses clock skew induced latency

Main Contributions

1. Constant Metadata

Hybrid Stable Time (HST)

A novel dependency tracking and stabilization protocol

2. Low Latency

Hybrid Logical/ Physical Clocks

Loose synchronization of physical clocks without suffering from clock skew

Hybrid Stable Time (HST)

- Only two scalar timestamps

Local Dependency Time

1

Tracks the dependencies on local items

Remote Stable Time

2

Summarizes dependencies on remote items

Existing dependency tracking methods

Dependency vectors

Cure[ICDCS'16]

+ Fresh, non-blocking snapshot

− Metadata size O(#DCs)

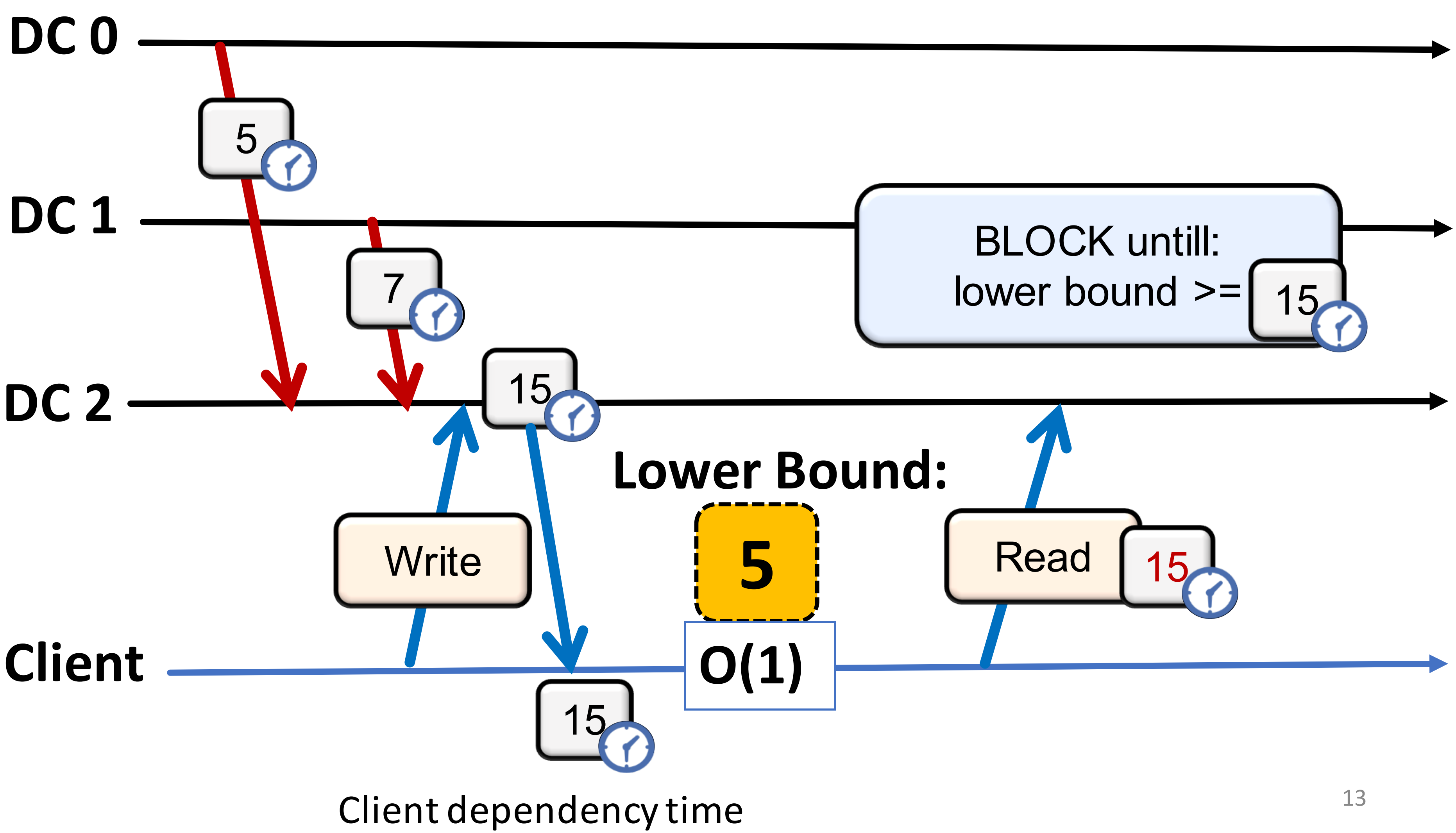
Single timestamp

GentleRain[SOCC'14]

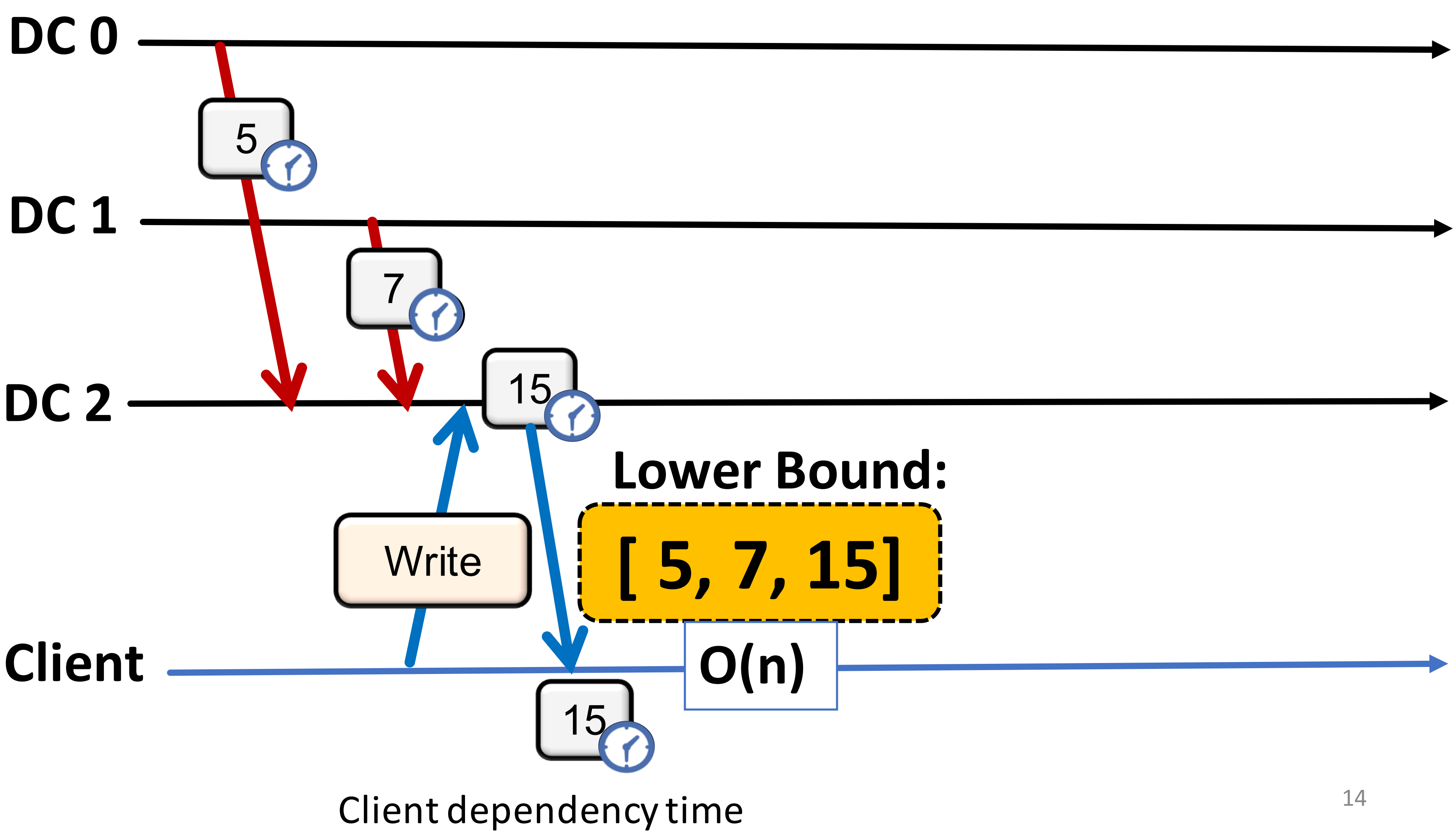
+ Metadata size O(1)

− Inter DC sync to install snapshot

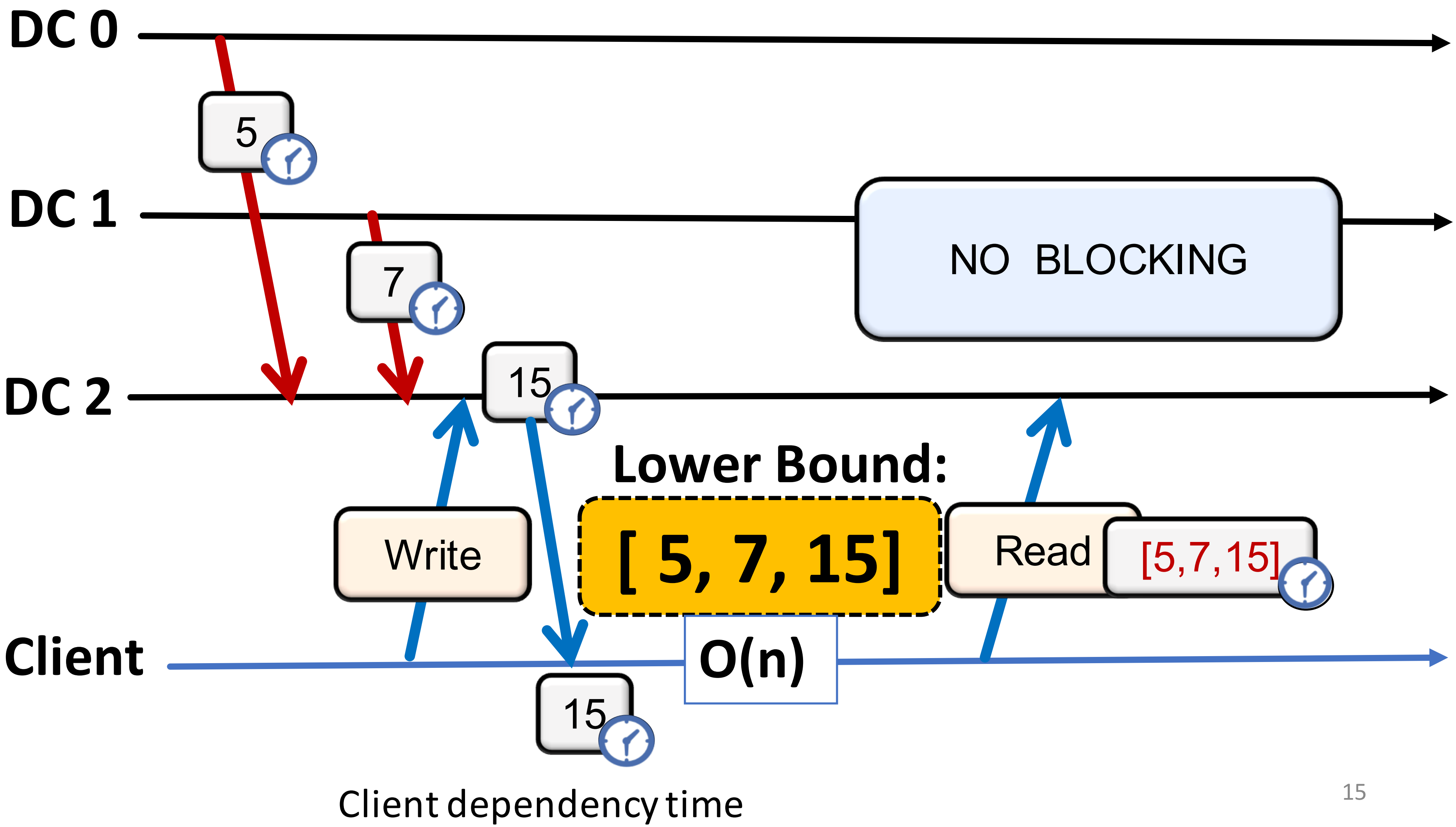
Single Scalar: GentleRain



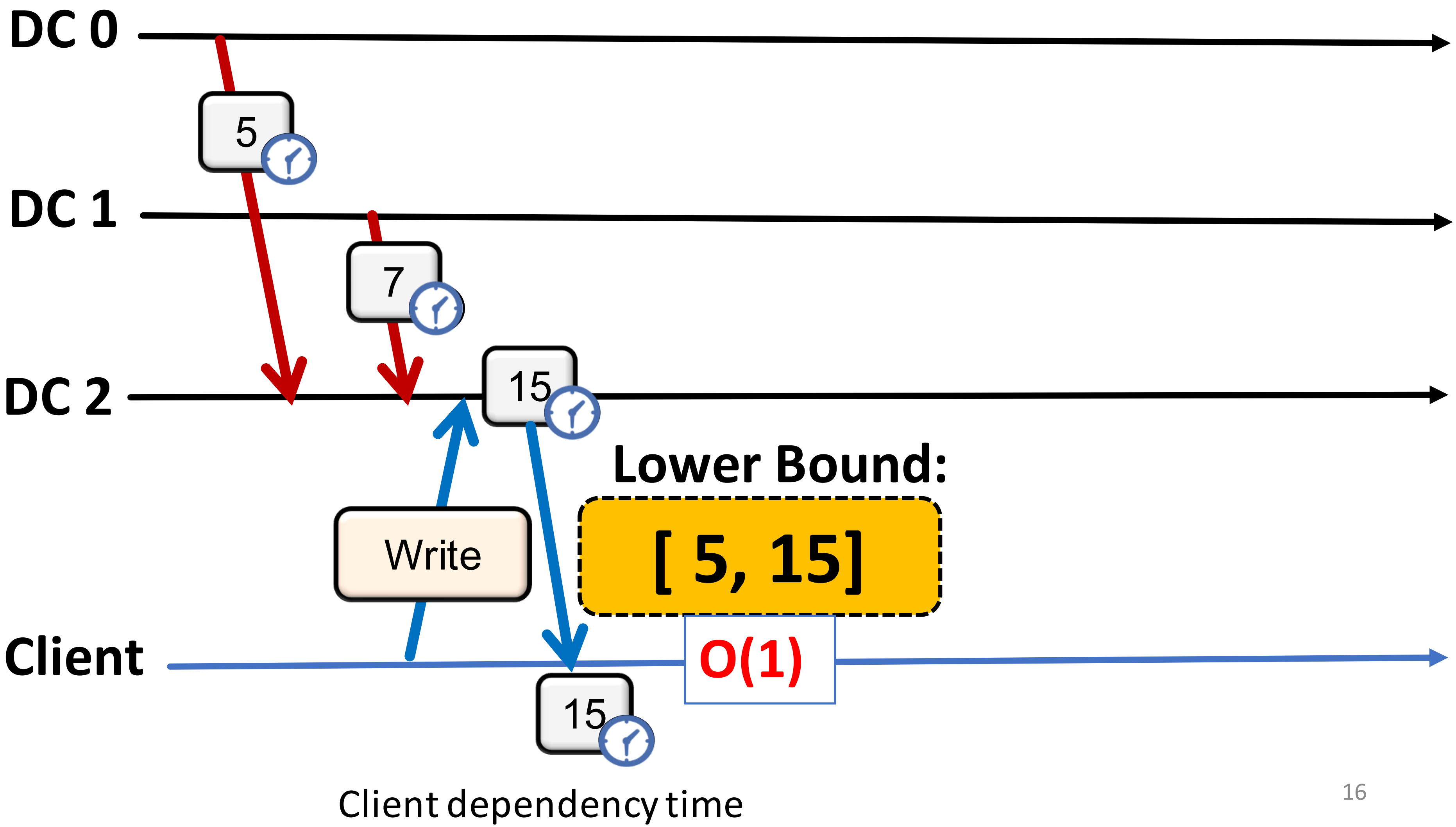
Dependency vectors: Cure



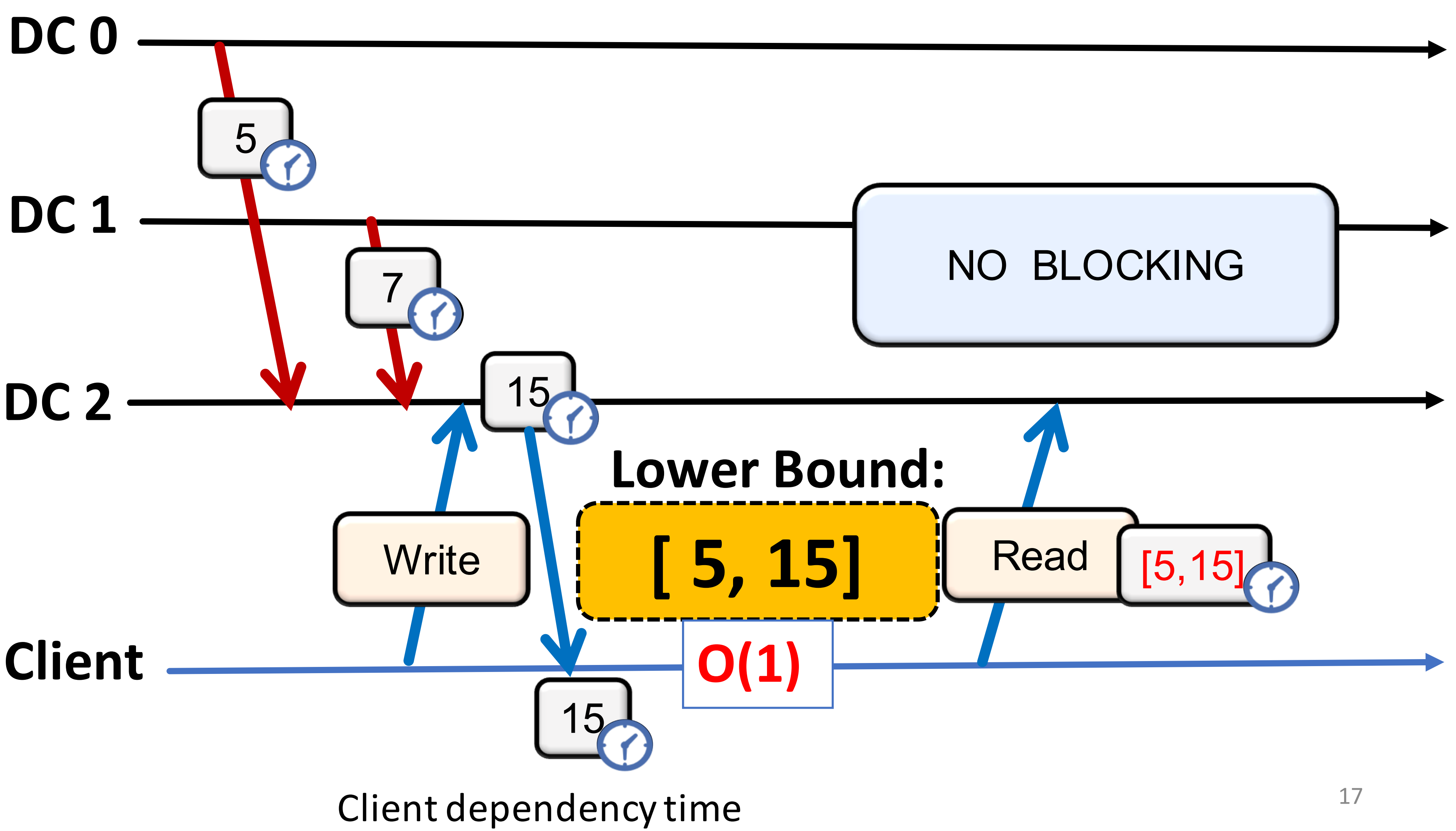
Dependency vectors: Cure



Local and Remote Time: Wren



Local and Remote Time: Wren



Hybrid Stable Time (HST)

- Only two scalar timestamps

Local Dependency Time

1

Tracks the dependencies on local items

Remote Stable Time

2

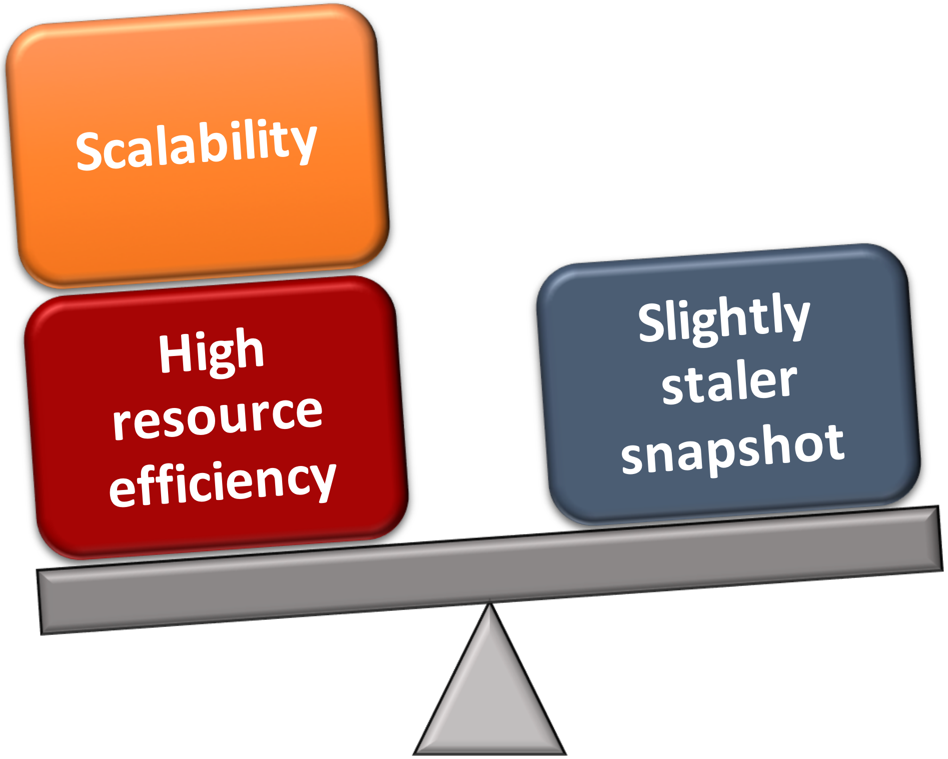
Summarizes dependencies on remote items

Remote Stable Time (RST)

- Computed periodically
- Lower bound on updates from remote DCs
- No additional inter DCs sync

HST Benefits & Trade-off

- Hits sweet spot in the meta-data size vs performance spectrum



Main Contributions

1. Constant Metadata

Hybrid Stable Time (HST)

A novel dependency tracking and stabilization protocol

2. Low Latency

Hybrid Logical/ Physical Clocks

Loose synchronization of physical clocks without suffering from clock skew

Existing update timestamping methods

Physical Clocks PC

+ (Loose) clock synchr. enables efficient dependency tracking

– Clock skew introduces uncertainty (latencies)

Logical Clocks LC

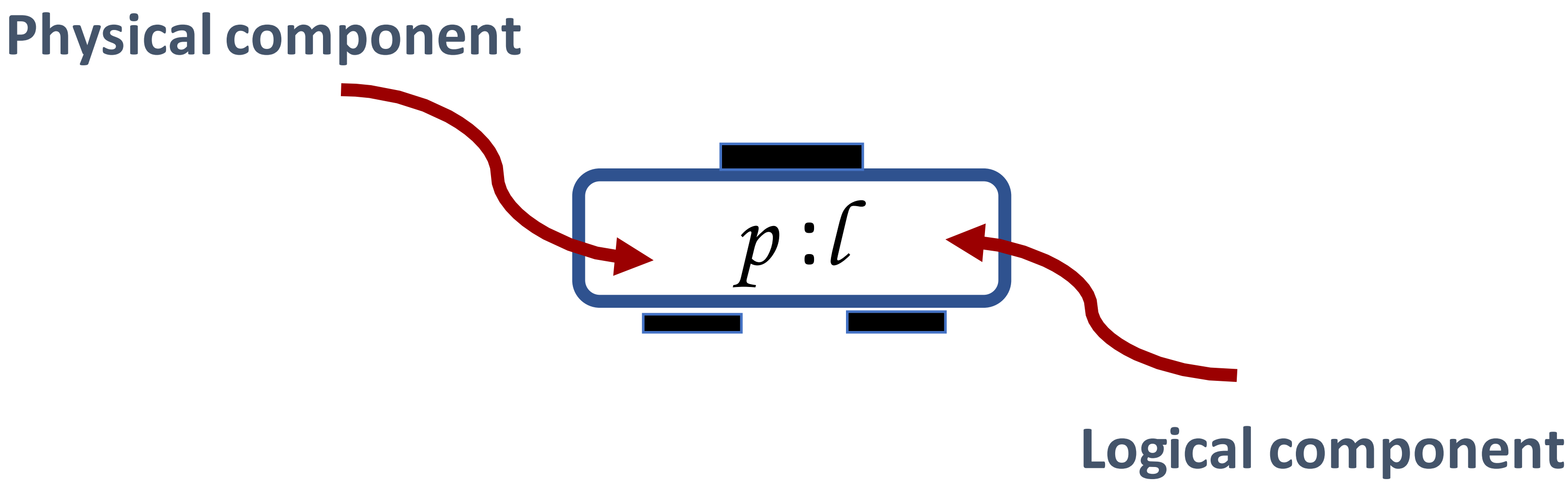
+ Capture dependency among events easily

– No syhncronization makes dependency tracking more costly

Hybrid Logical/Physical clocks (HCL)

- *Logical Physical Clocks*, OPODIS'14
- Best of both worlds
  - Captures the causality relationship
  - Inherits loose synchronization of PC
  - No clock skew as LC

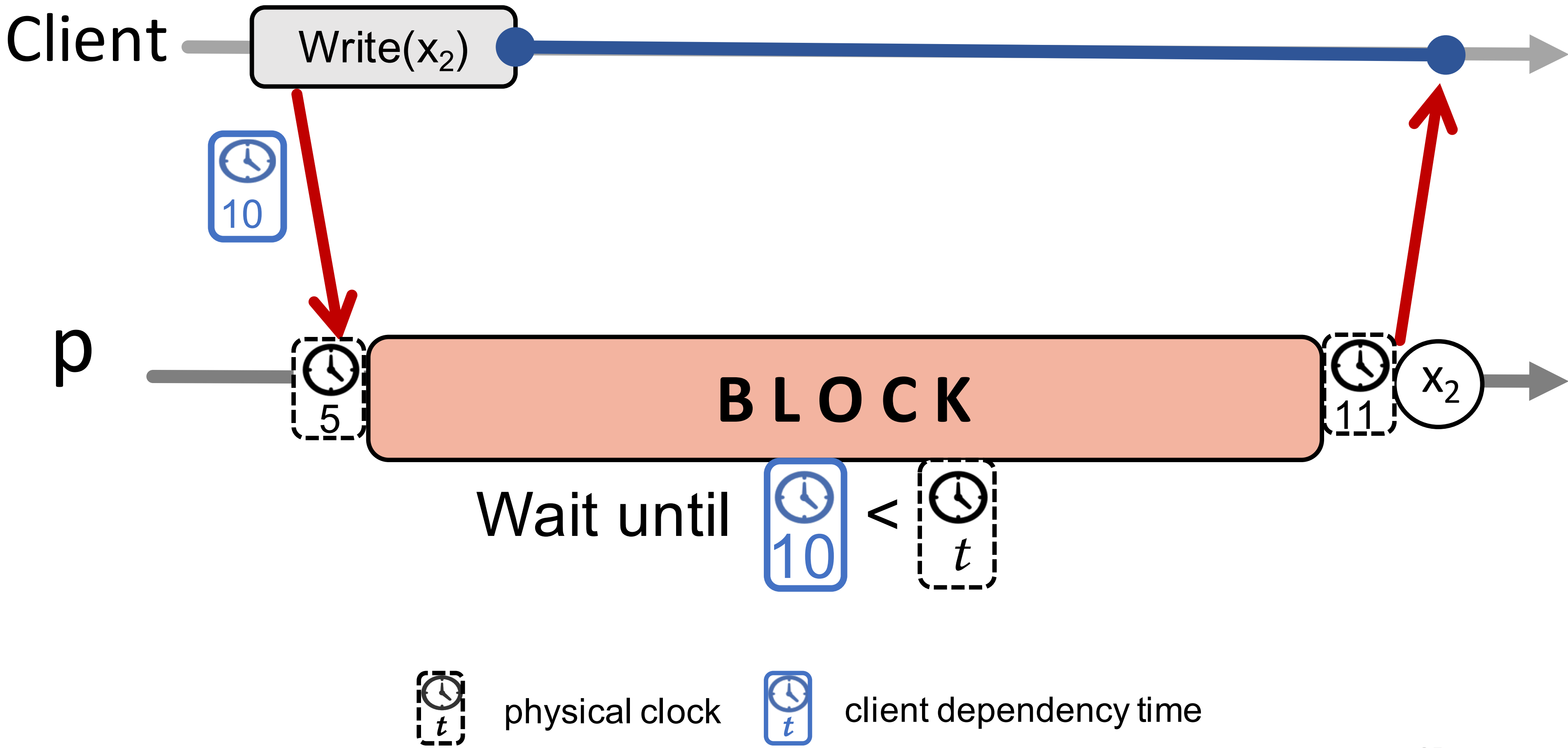
Hybrid Logical/Physical clocks (HCL)



Clock skew problem

Invariant: timestamps must reflect causality

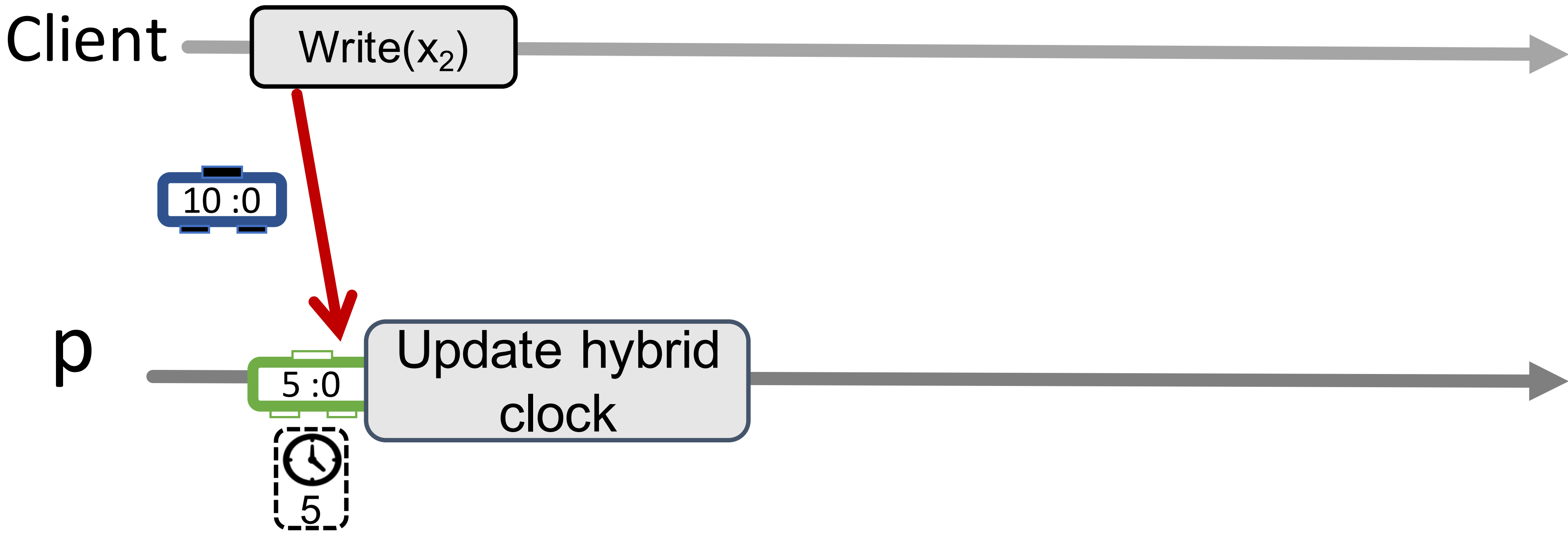
Client's dep. time < x2's update time



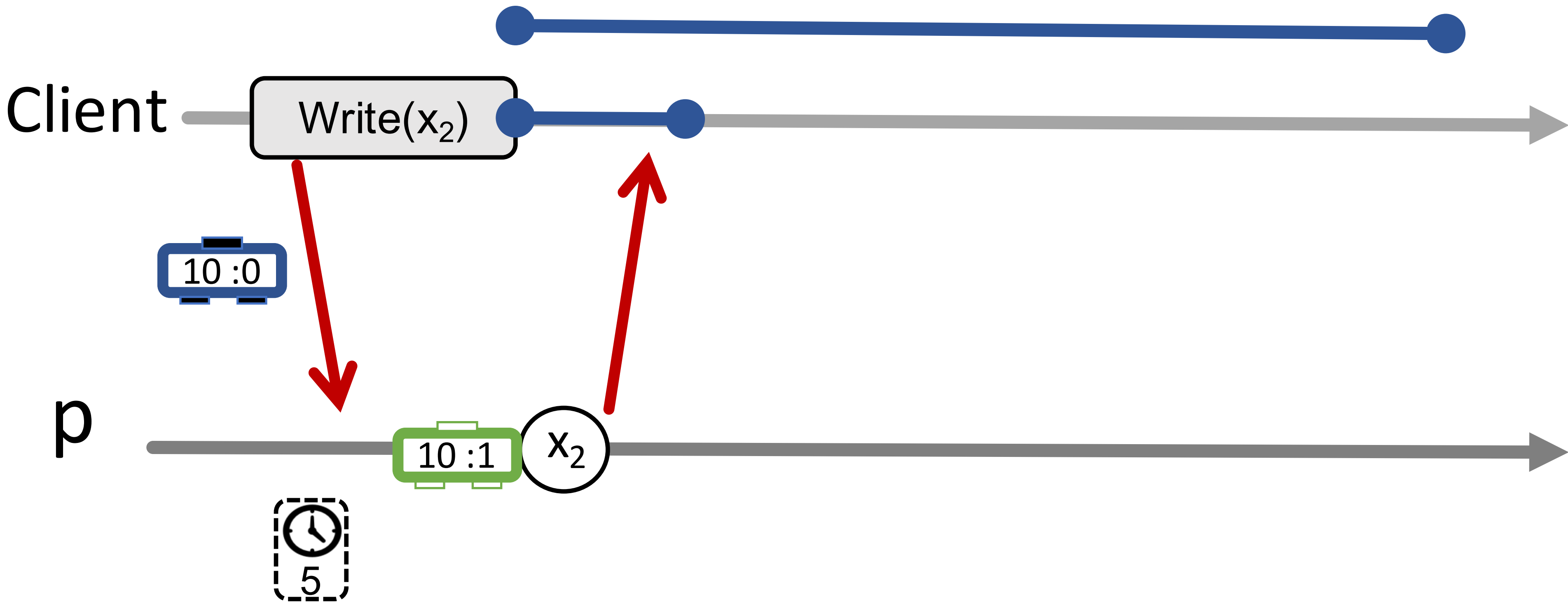
Clock skew solution

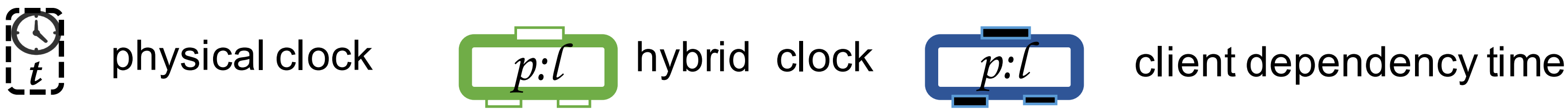
Invariant: timestamps must reflect causality

Client's dep. time < x2's update time

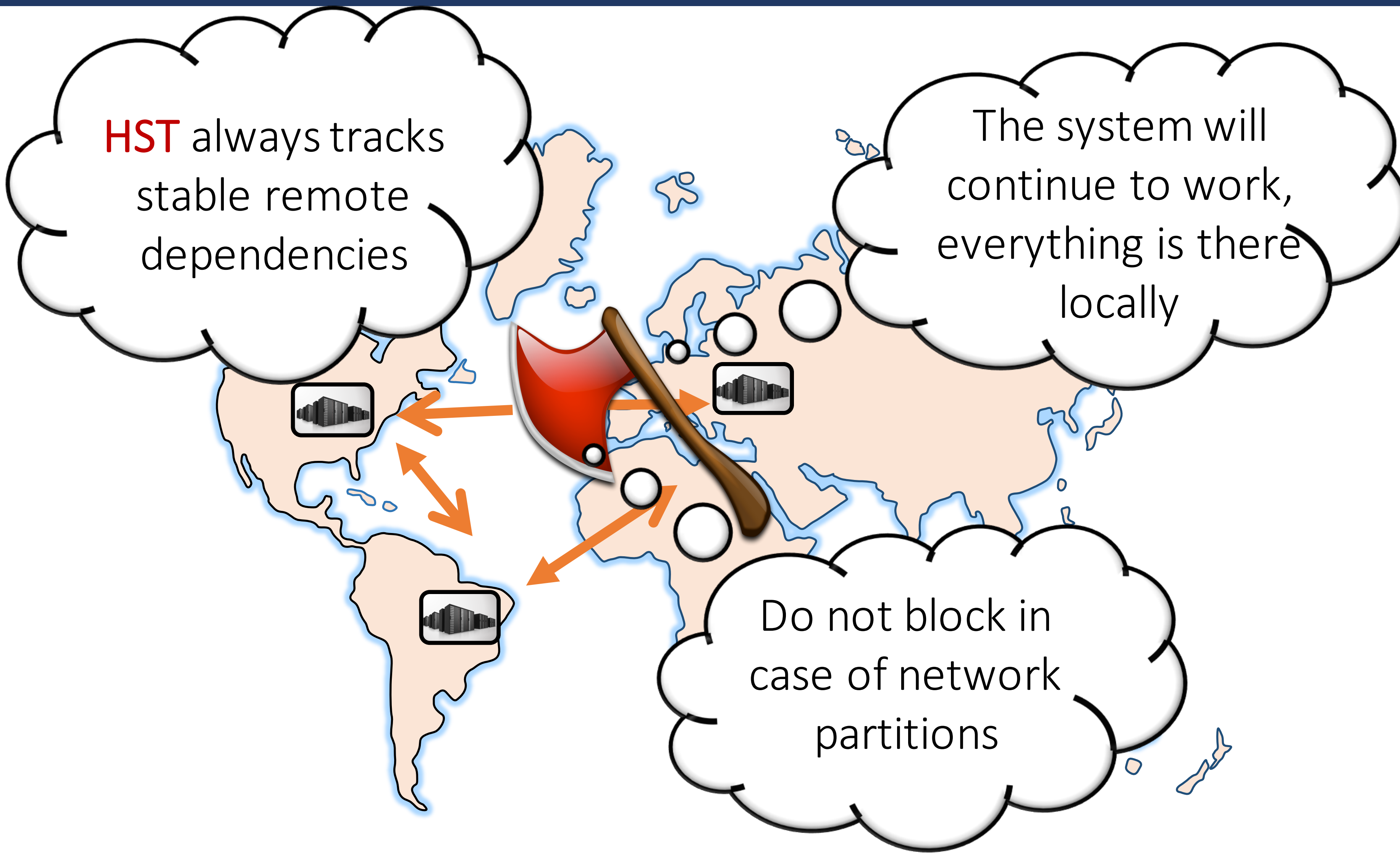


Clock skew solution





Availability



Outline

- Introduction
- Our work- Wren
- Related work
- Summary

Related Systems

	Low Latency	Metadata	Availability (Transact.)	Snapshot freshness
Cure [ICDCS'16]	Clock skew	#DCs	Yes	Higher
GentleRain [SOCC'14]	Clock skew Inter-DC syn	O(1)	No	Lower
Occult [NSDI'17]	Inter DC sync	O(#DCs)	No	Highest
Wren	Yes	O(1)	Yes	Lower

Our work: Wren

The first transactional causally consistent geo-replicated system that at the same time has:

- Constant metadata
- Low latency
- Always-available

Summary

- Wren: The first transactional causally consistent geo-replicated system that at the same time has:
  - Constant metadata
    - Hybrid Stable Time
  - Low latency
    - Hybrid Logical/Physical Clocks
  - Always-available

Thank you!