Shadow: The Case for Determinism on the Edge

Matthew Furlong, Andrew Quinn, and Jason Flinn

University of Michigan

Concerns for the Edge

Edge Concern	Effect	Problem

Concerns for the Edge

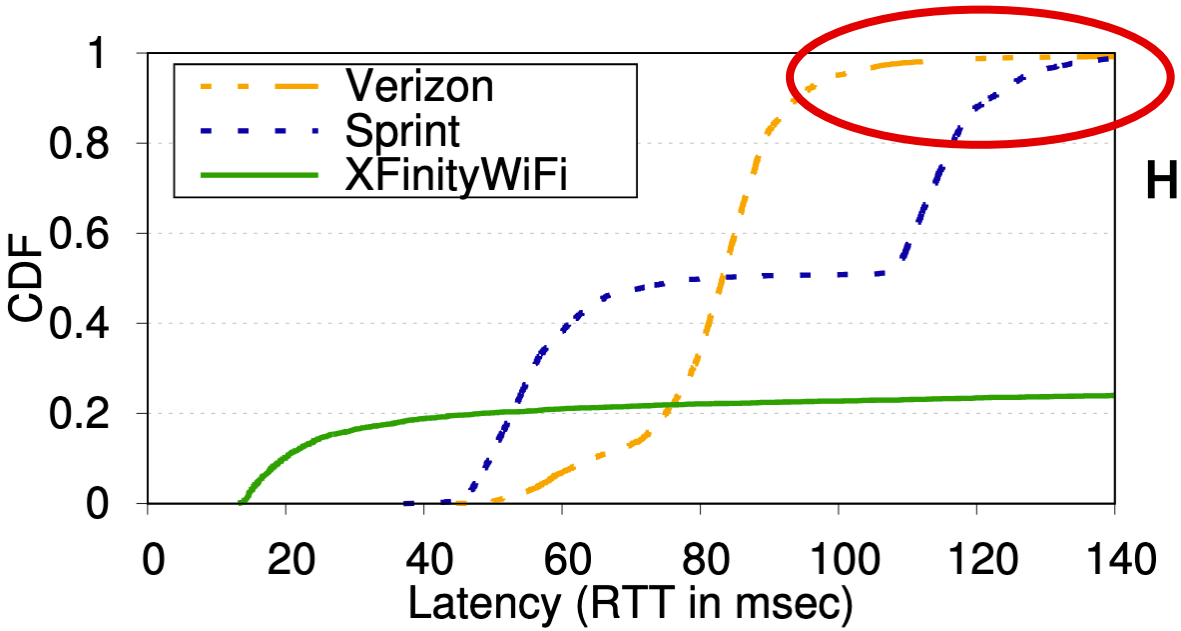
Edge Concern	Effect	Problem
Provisioning	More jitter	

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability



High tail latency!

[1] Lee, H., Flinn, J., And Tonshal, B. Raven: Improving interactive latency for the connected car.

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost
Reliability	More faults	

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost
Reliability	More faults	• Numerous providers

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost
Reliability	More faults	• Numerous providers • Distributed

Concerns for the Edge

Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost
Reliability	More faults	• Numerous providers • Distributed
More data sources	Privacy concerns	

Concerns for the Edge

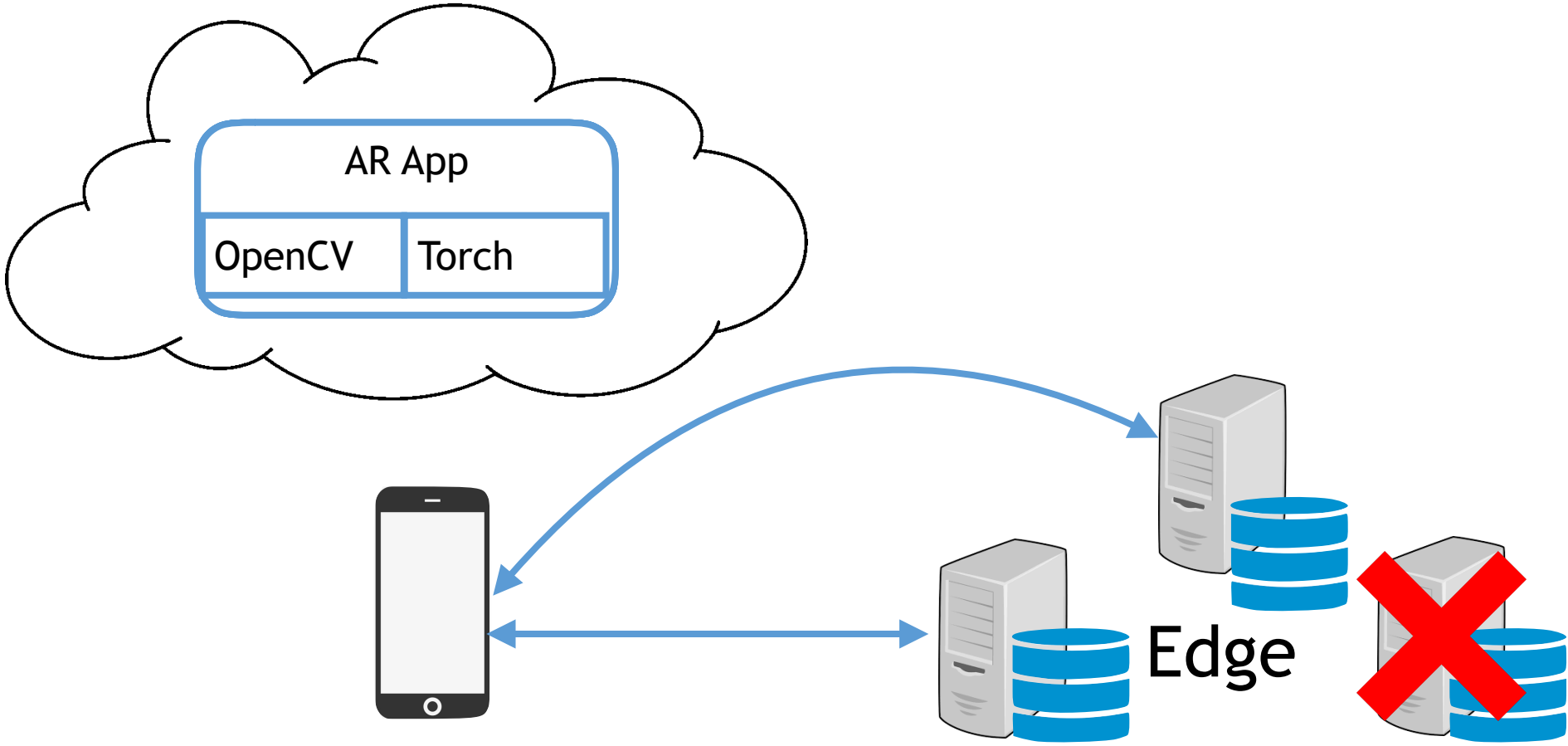
Edge Concern	Effect	Problem
Provisioning	More jitter	• Limited edge resources • Mobile network variability
Mobility	Performance variability	• Low-latency connections are lost
Reliability	More faults	• Numerous providers • Distributed
More data sources	Privacy concerns	• More IoT devices = more sensitive user data

Moving From the Cloud to Edge

- Applications are typically latency-sensitive or require high bandwidth

Moving From the Cloud to Edge

- Applications are typically latency-sensitive or require high bandwidth

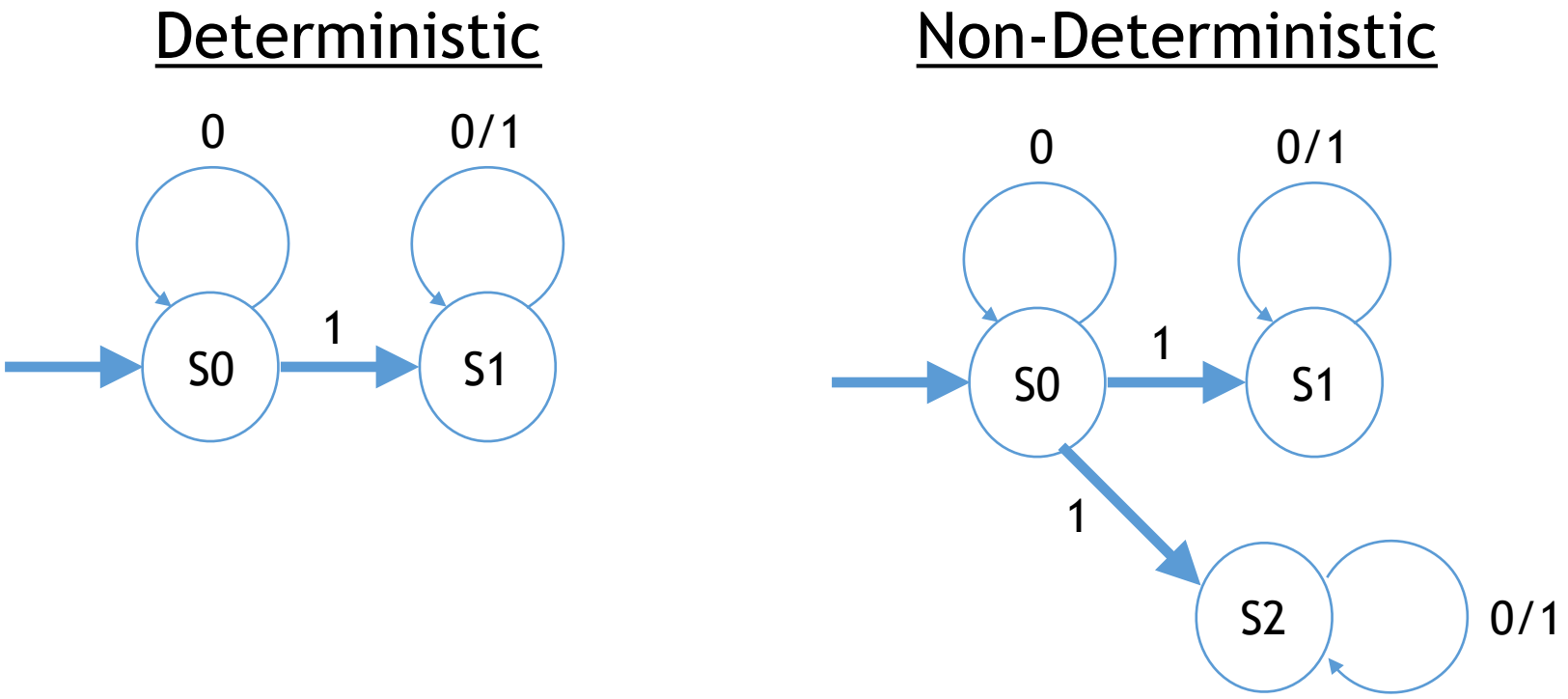


Outline

- Motivation
- Determinism to the rescue
- Shadow: Deterministic containers
- Evaluation

Determinism to the Rescue

- A deterministic execution of an application provides the necessary support for edge systems



Determinism to the Rescue

- Determinism allows applications to be replicated across multiple servers

Determinism to the Rescue

- Determinism allows applications to be replicated across multiple servers
- Determinism enables migration with minimal downtime

Migration

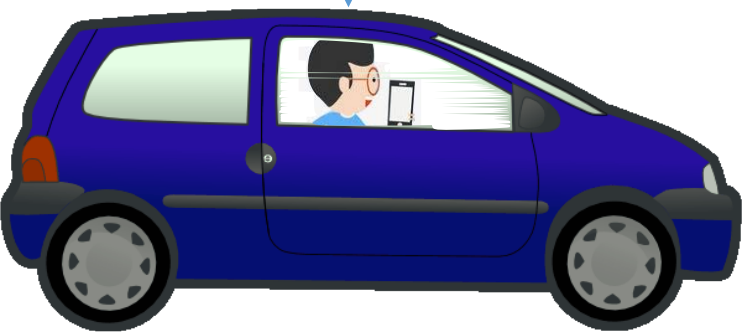
Server inputs: a







Low-latency connection



Server Responses: 1

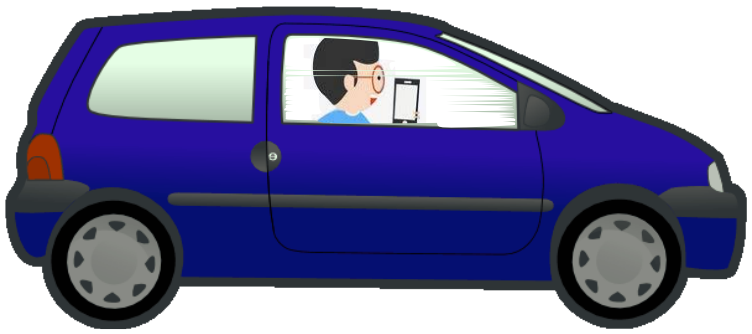
Migration

Server inputs: a





Better connection discovered



Server Responses: 1

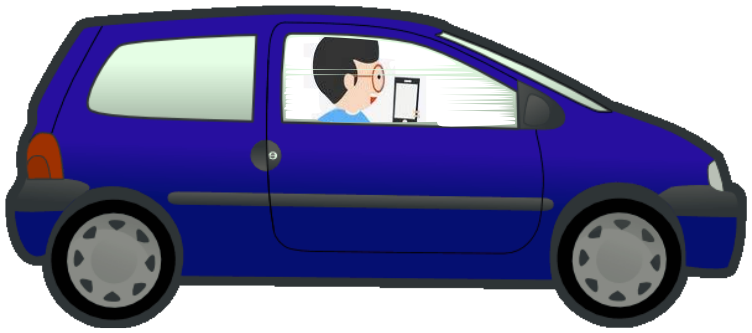
Migration

Server inputs: a



Initiate migration!





Server Responses: 1

Migration

Server inputs: a b

Server inputs: a

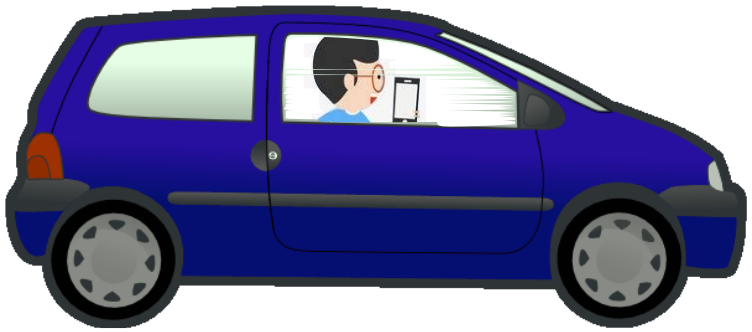


Checkpoint Application



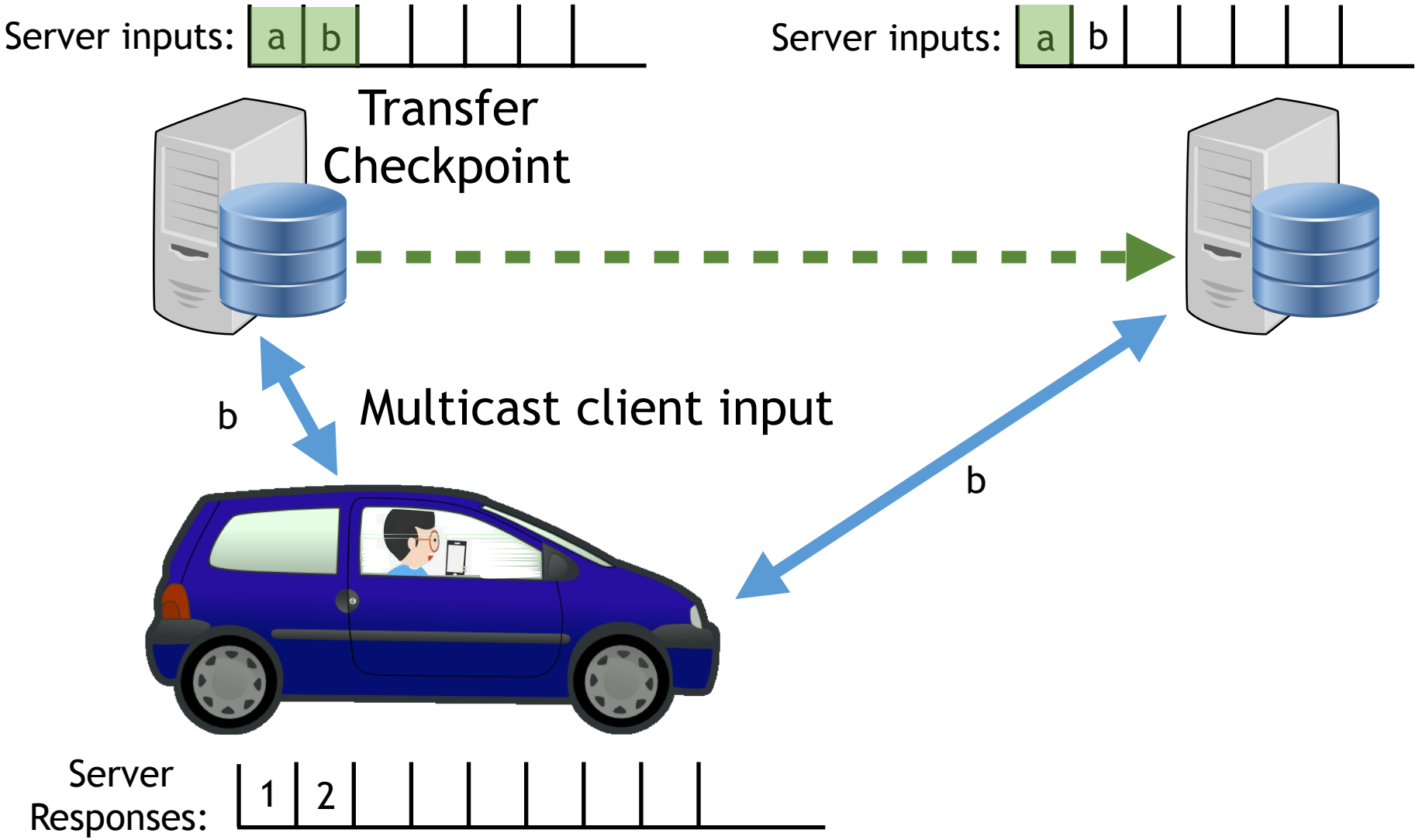


b

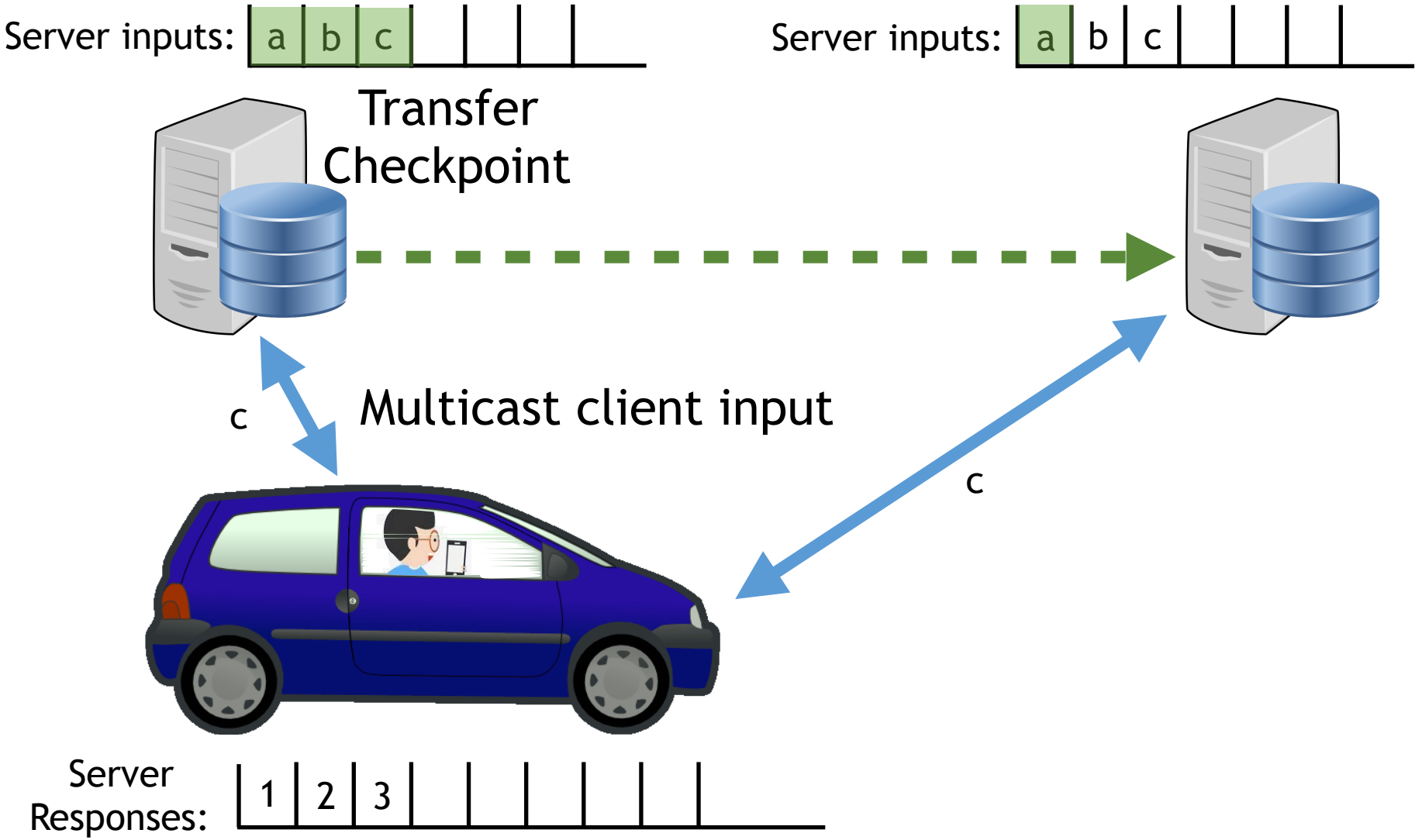


Server Responses: 1 ?

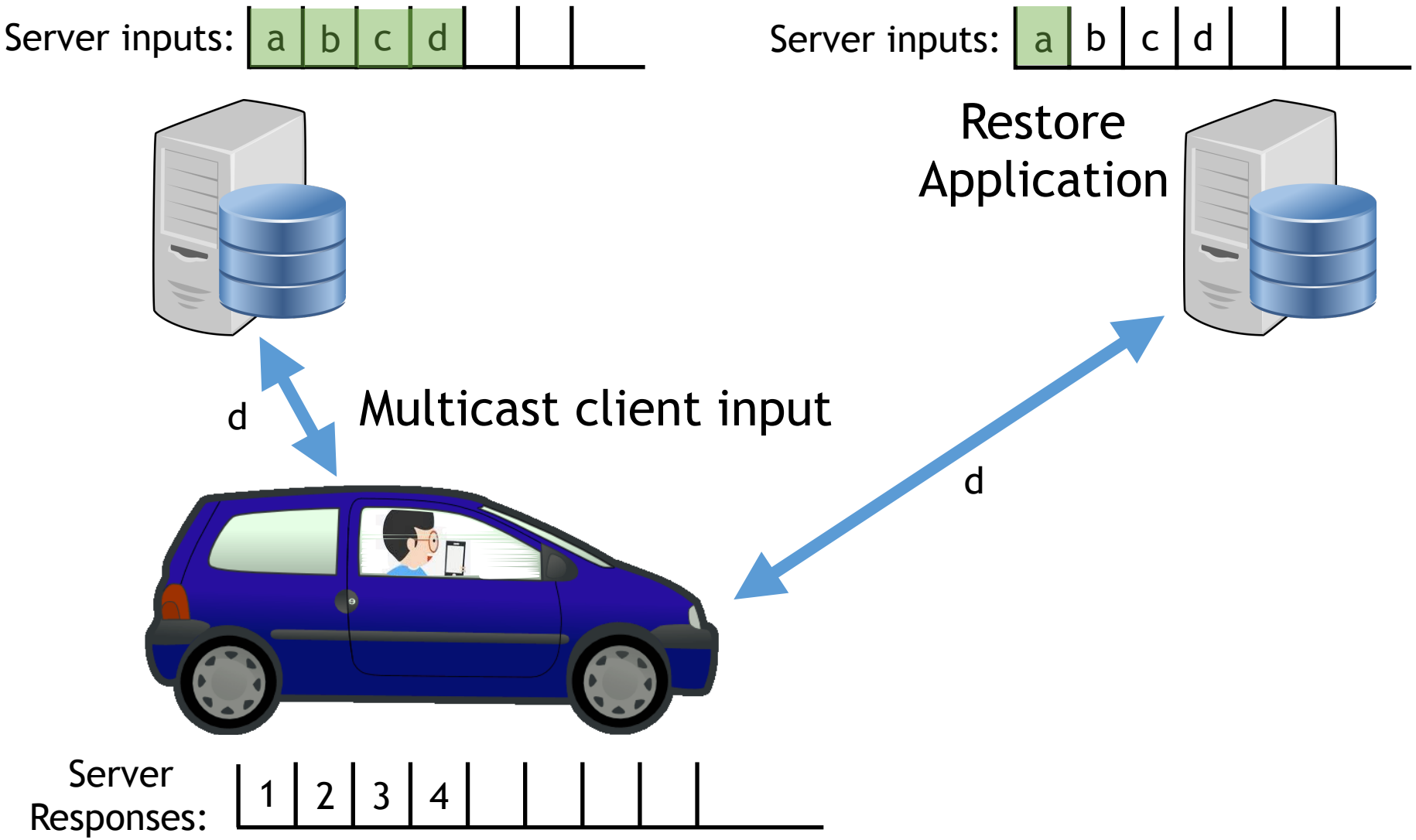
Migration


Server inputs:
a
b
Server inputs:
a
b
Transfer
Checkpoint
b
Multicast client input
b
Server
Responses:
1
2

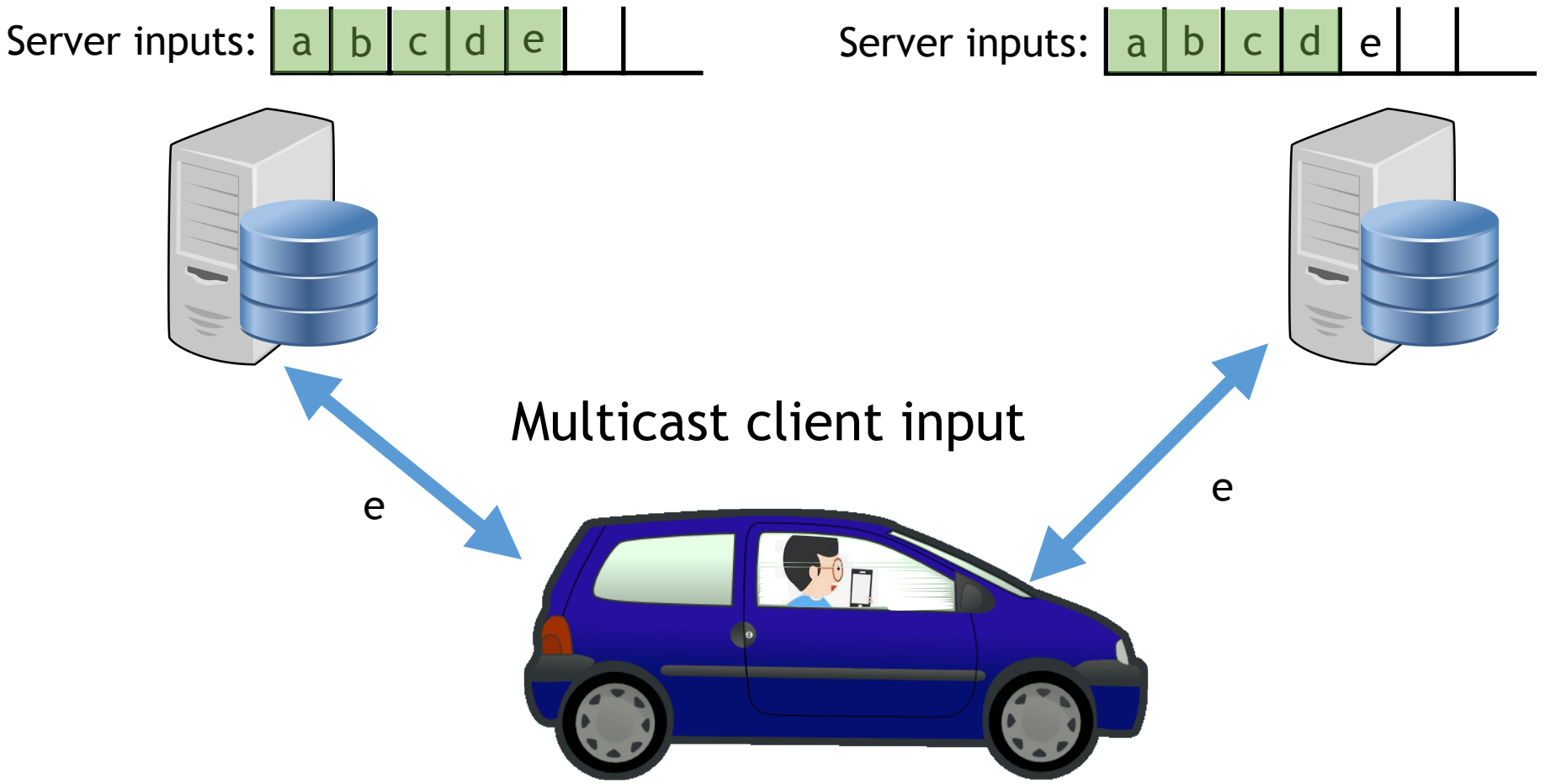
Migration


Server inputs:
a
b
c
Server inputs:
a
b
c
Transfer
Checkpoint
c
Multicast client input
c
Server
Responses:
1
2
3

Migration


Server inputs:
a
b
c
d
Server inputs:
a
b
c
d
Restore Application
d
Multicast client input
d
Server Responses:
1
2
3
4

Migration


Server inputs: a b c d e
Server inputs: a b c d e
Multicast client input
e
e
Server Responses: 1 2 3 4 5

Migration

Server inputs: a b c d e f

Server inputs: a b c d e f

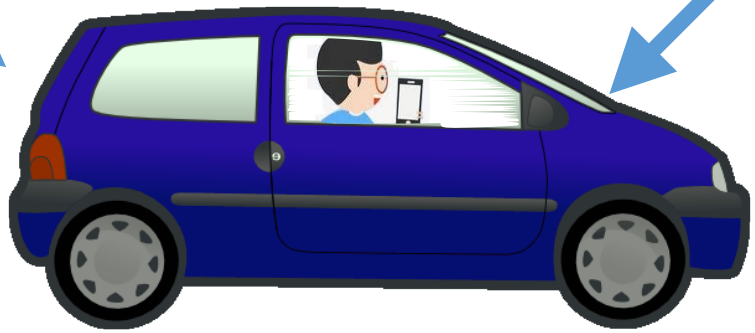




Multicast client input

f

f



Server Responses: 1 2 3 4 5 6

Migration

Server inputs: a b c d e f

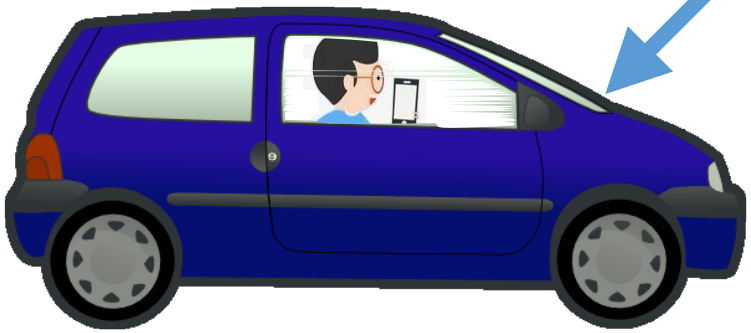


Server inputs: a b c d e f



Migration complete!

g



Client inputs: 1 2 3 4 5 6

Determinism to the Rescue

Edge Concern	Problem	Solution
Provisioning	• Limited edge resources • Mobile network variability	**Determinism**
Mobility	• Low-latency connections are lost	**Determinism**
Reliability	• Numerous providers • Distributed	**Determinism**
More data sources	• More IoT devices = more sensitive user data	**Determinism**

Determinism to the Rescue

Edge Concern	Problem	Solution
Provisioning	• Limited edge resources • Mobile network variability	**Determinism**
Reliability	• Numerous providers • Distributed	**Determinism**
More data sources	• More IoT devices = more sensitive user data	**Determinism**

Challenge: Applications are inherently non-deterministic!

Determinism to the Rescue

- Non-determinism in applications
  - Randomness
  - Time
  - Multi-threading
  - Network communication
  - Inter-process communication (IPC)

Determinism to the Rescue

- Non-determinism in applications
  - Randomness
  - Time
  - Multi-threading
  - Network communication
  - Inter-process communication (IPC)
- Challenge: How can we make non-deterministic applications behave deterministically??

Outline

- Motivation
- Determinism to the rescue
- Shadow: Deterministic containers
- Evaluation

Shadow

- Goal: Eliminate sources of non-determinism

Shadow

- Goal: Eliminate sources of non-determinism
  - Minimize application modifications

Shadow

- Goal: Eliminate sources of non-determinism
  - Minimize application modifications
  - Maintain similar performance to non-deterministic executions

Deterministic Containers

- Shadow removes non-determinism from applications deployed within containers
  - Multi-threading
  - OS Events
  - Client inputs
  - External Sources of Non-Determinism

Deterministic Thread Scheduler

- Shadow uses a deterministic thread scheduler based on *weak* determinism[1]

[1] Olseweski, M., Ansel, J., And Amarsinghe, S. Kendo: Efficient deterministic multithreading in software.

Deterministic Thread Scheduler

- Shadow uses a deterministic thread scheduler based on *weak* determinism[1]
- Weak determinism globally orders all synchronization operations

[1] Olseweski, M., Ansel, J., And Amarsinghe, S. Kendo: Efficient deterministic multithreading in software.

Deterministic Containers

- Shadow removes non-determinism from applications deployed within containers
  - Multi-threading
  - OS Events
  - Client inputs
  - External Sources of Non-Determinism

Operating System Events

- System Calls
  - getrandom()
  - gettimeofday()
  - getpid()
  - …

Deterministic Containers

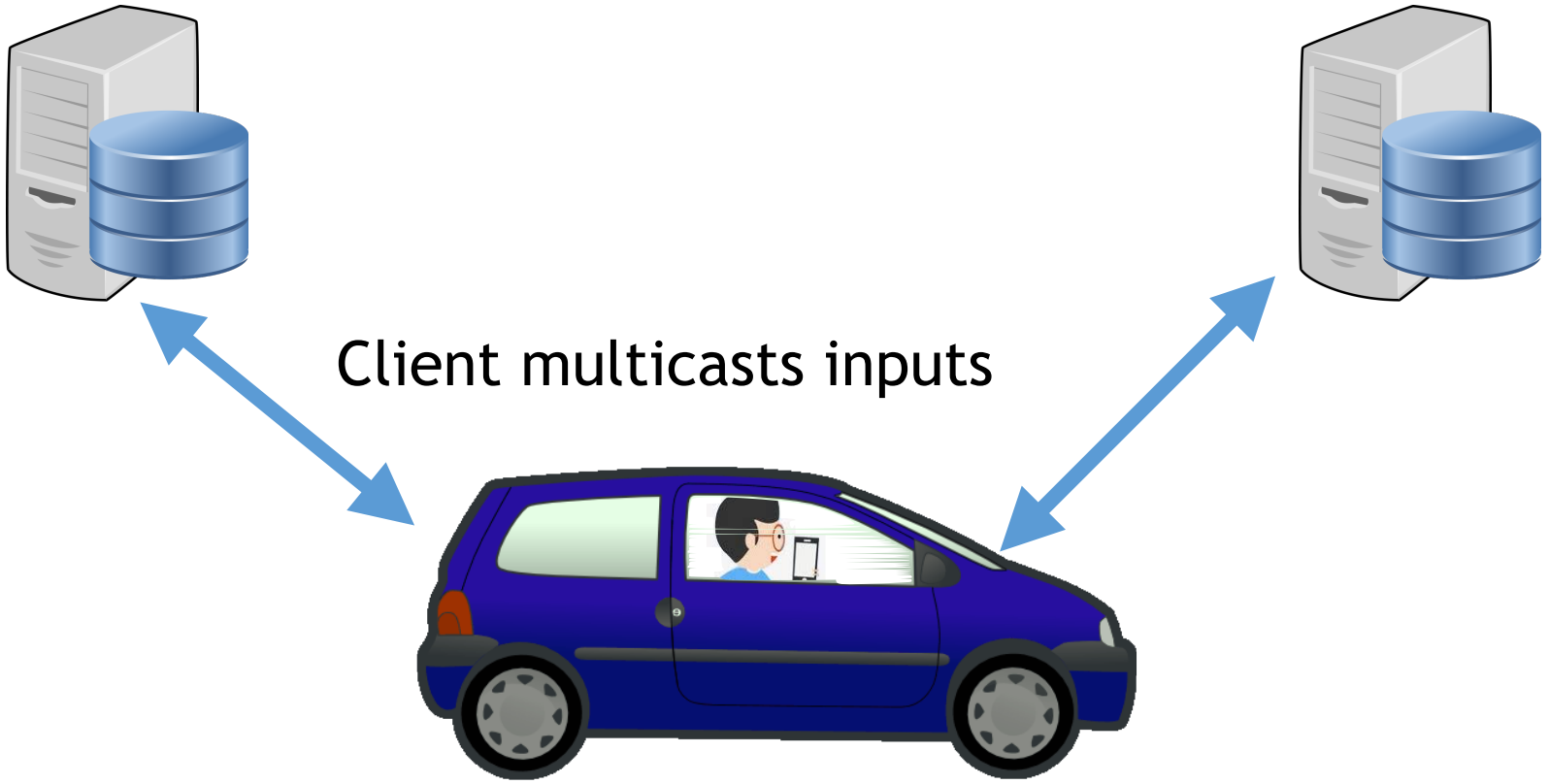
- Shadow removes non-determinism from applications deployed within containers
  - Multi-threading
  - OS Events
  - Client inputs
  - External Sources of Non-Determinism

Client Inputs

- Determinism is only useful if replicas receive the same sequence of inputs

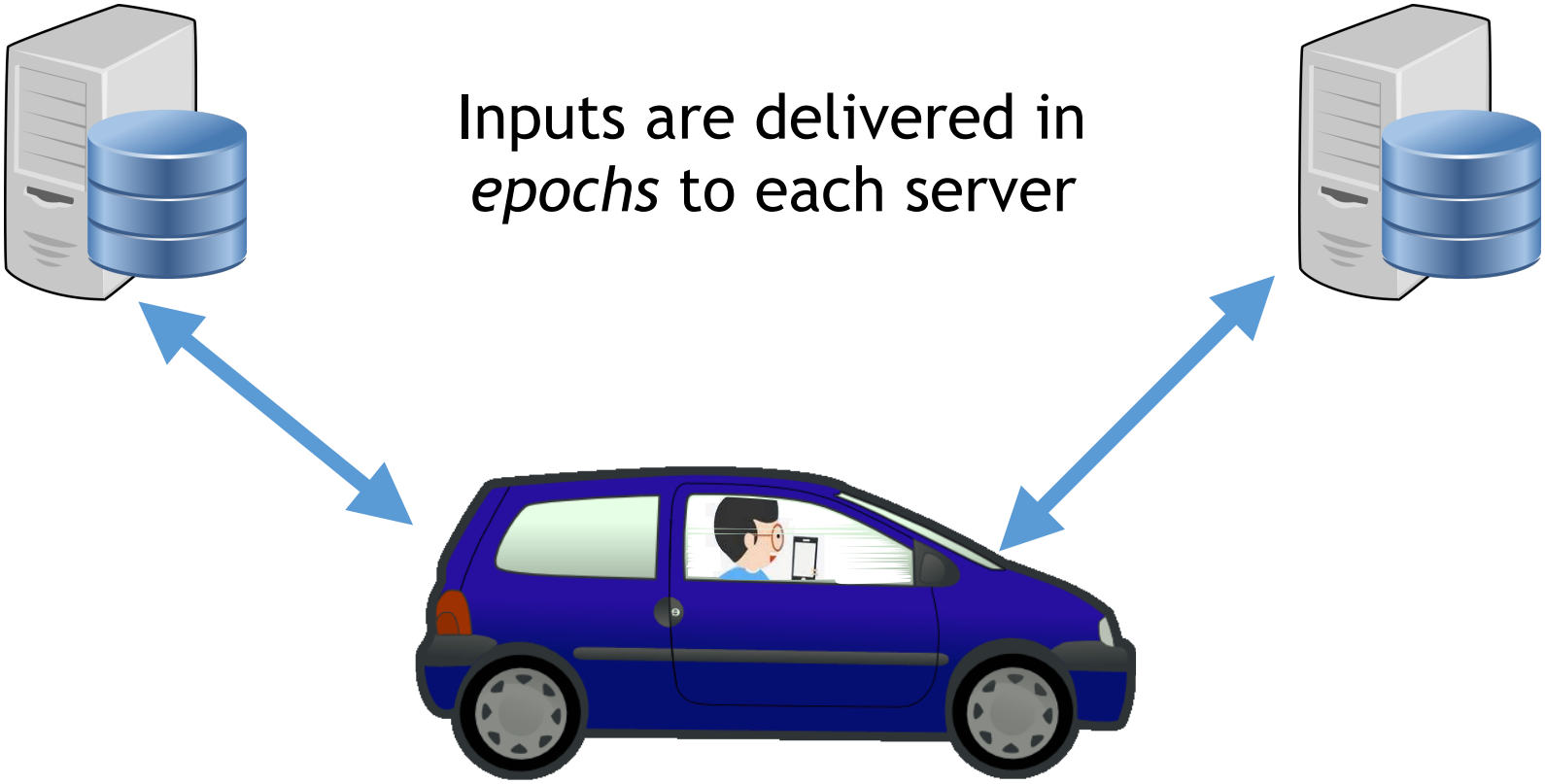
Client Inputs

- Determinism is only useful if replicas receive the same sequence of inputs



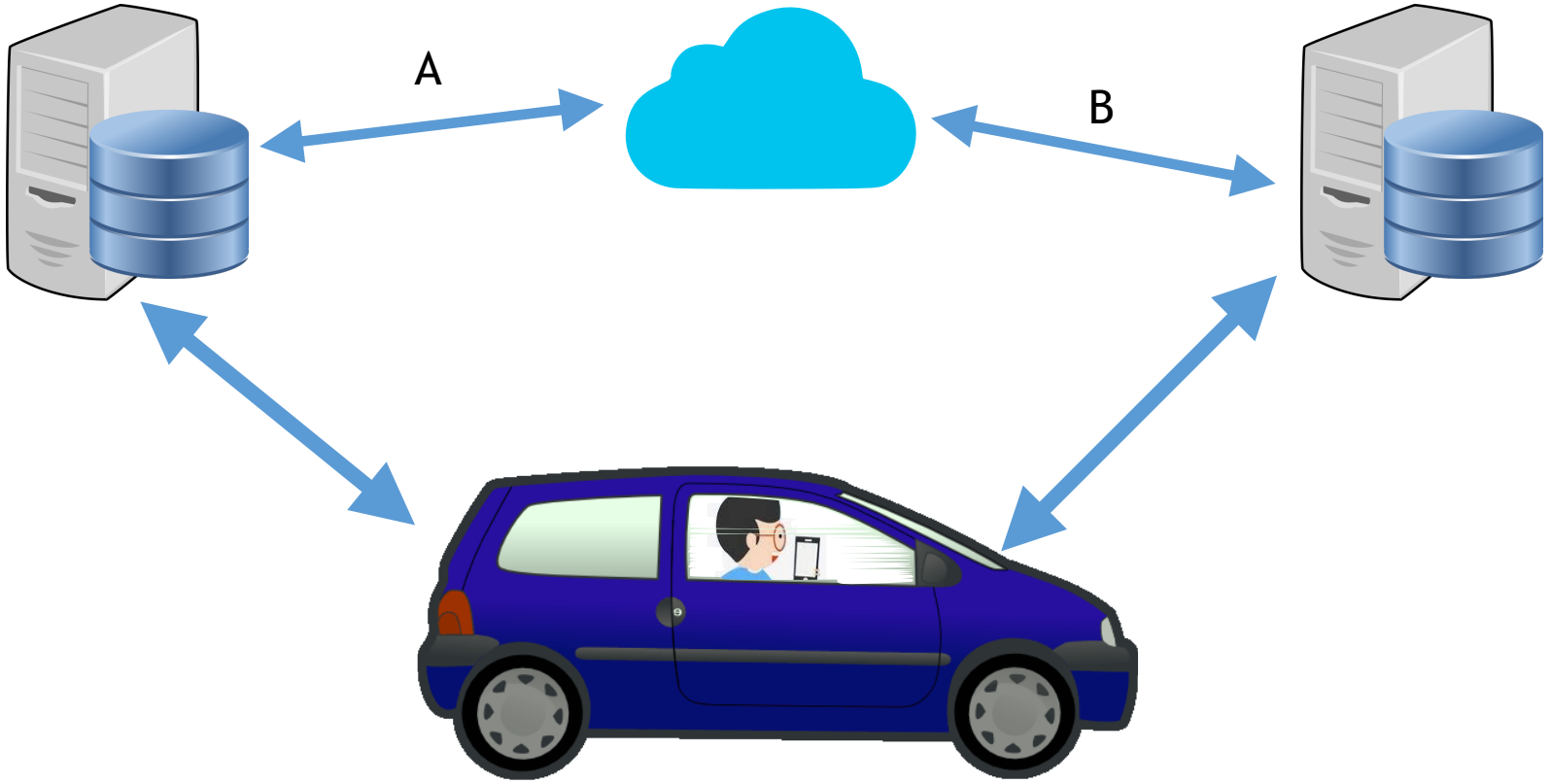
Client Inputs

- Determinism is only useful if replicas receive the same sequence of inputs



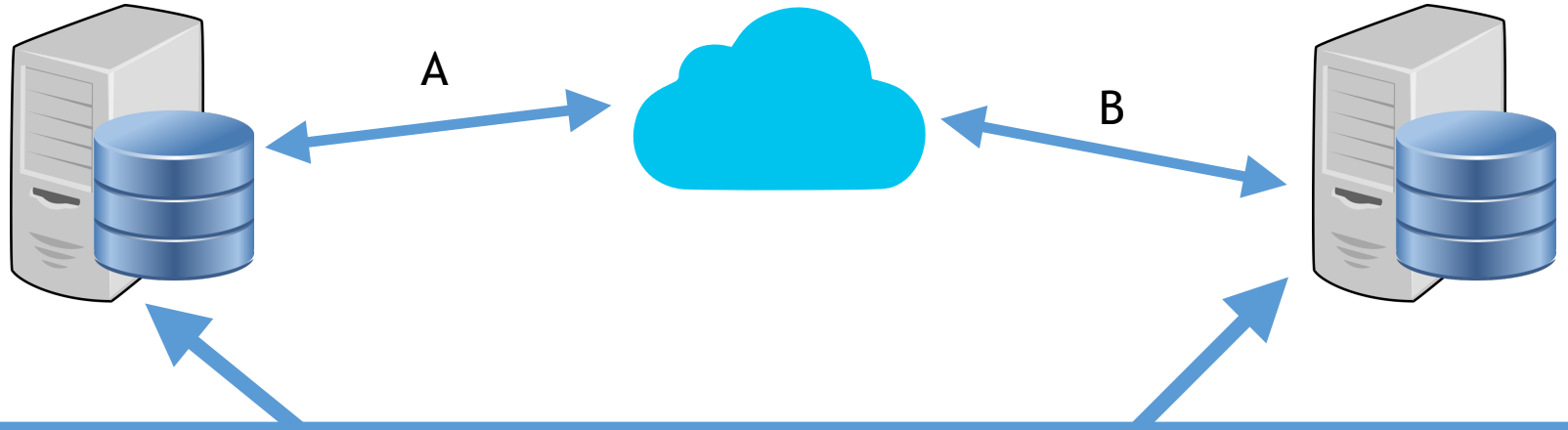
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



External Sources

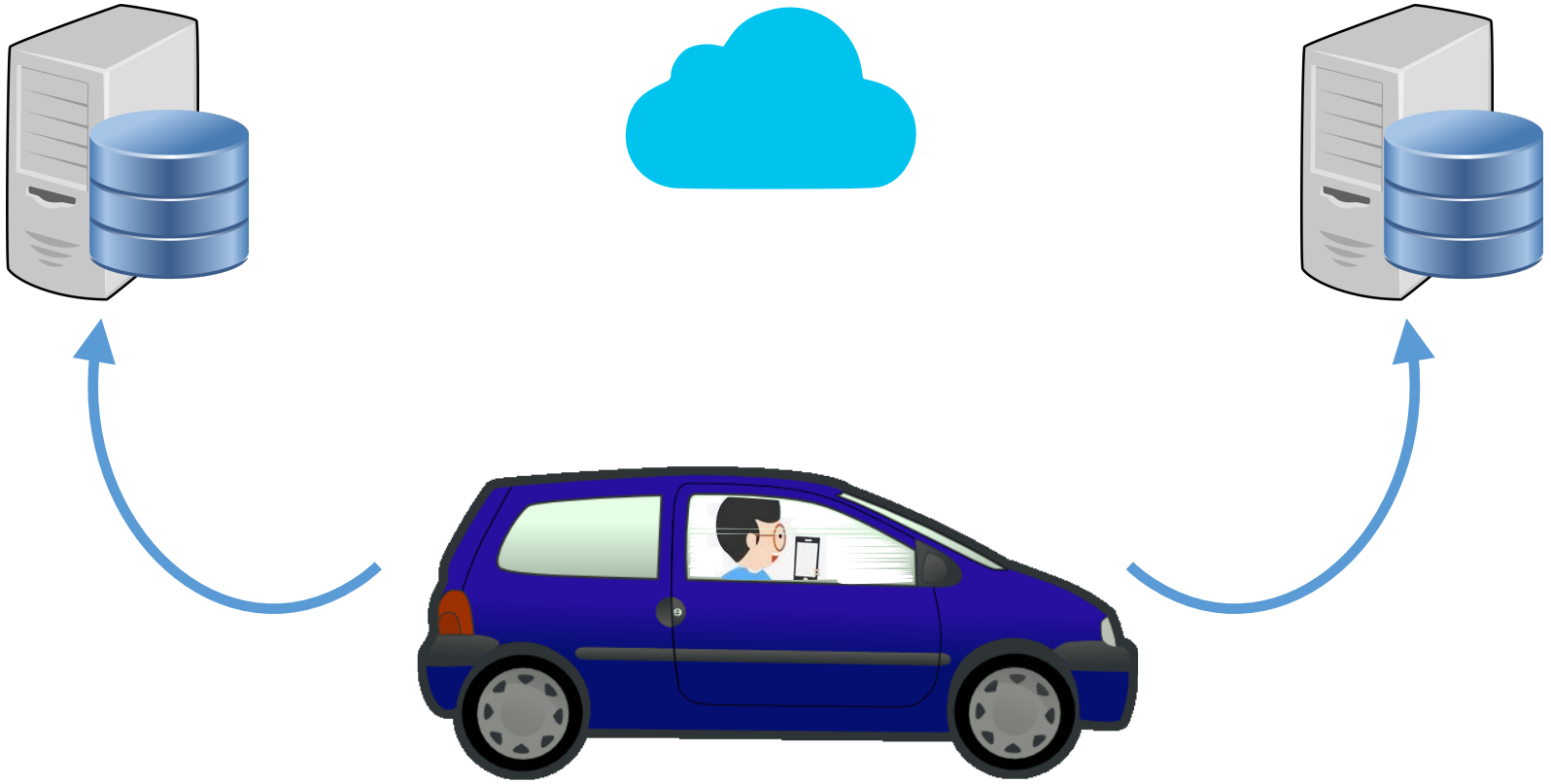
- Determinism is only useful if replicas receive the same sequence of inputs



3rd party service responses are not identical

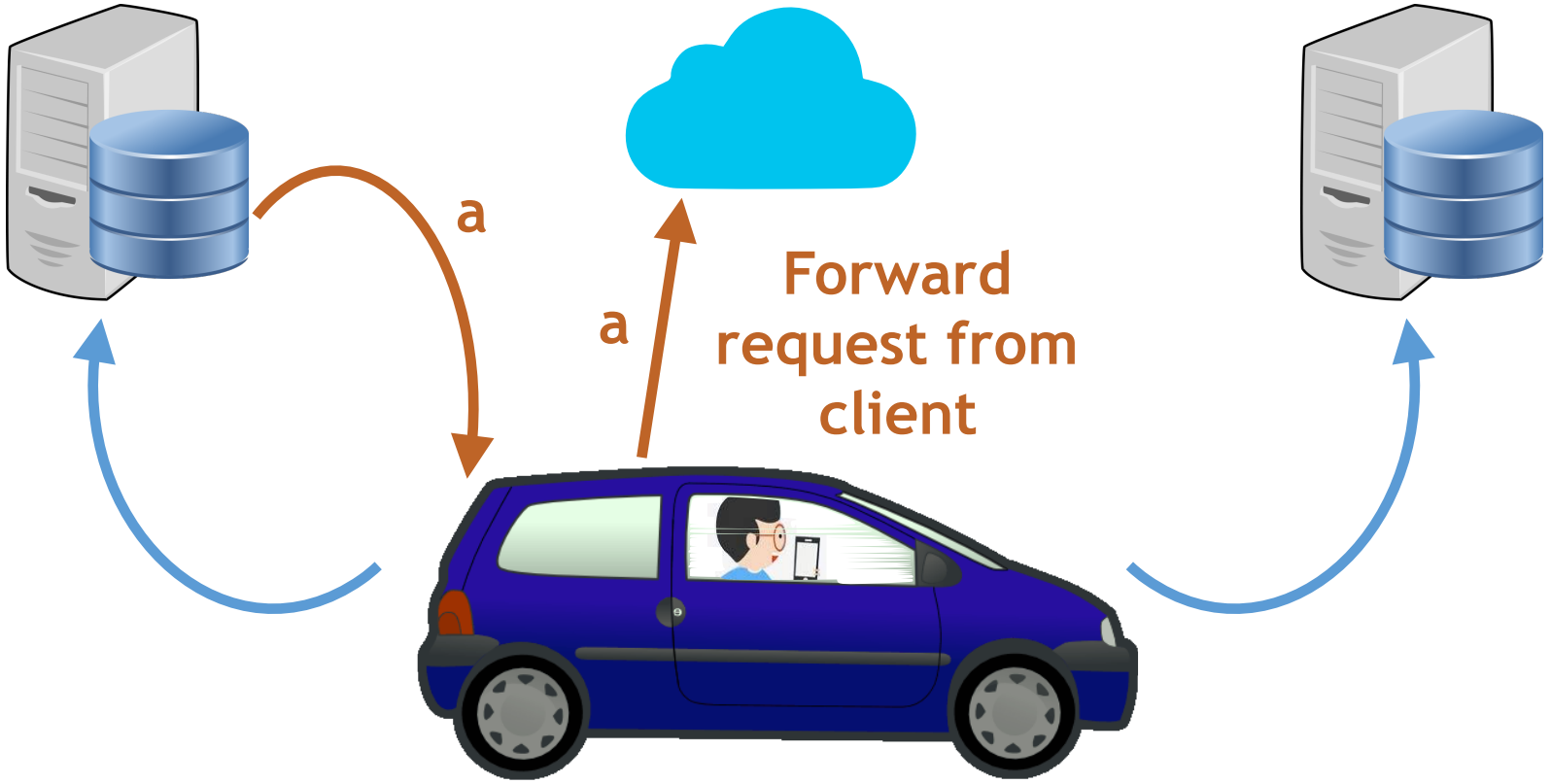
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



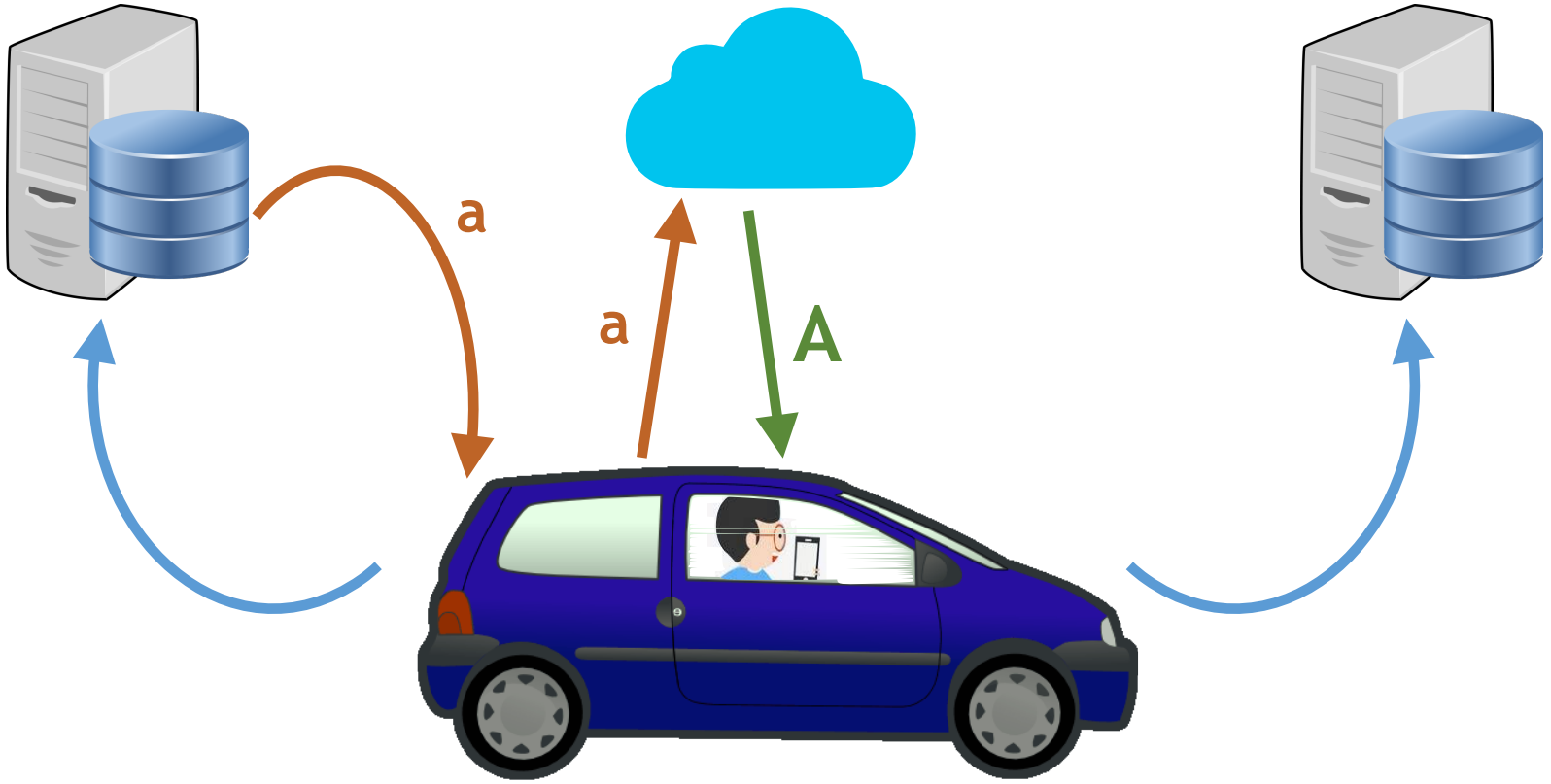
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



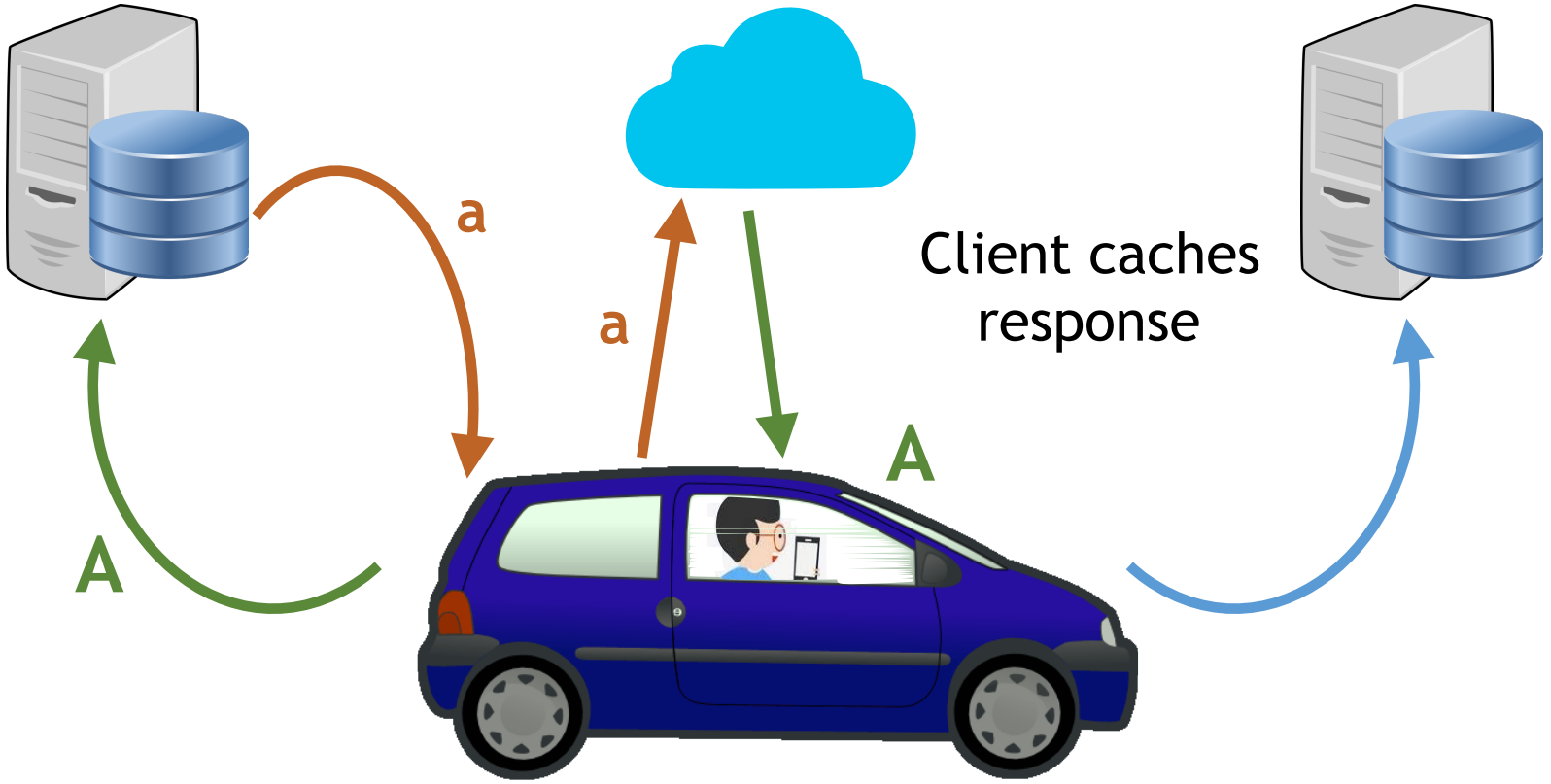
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



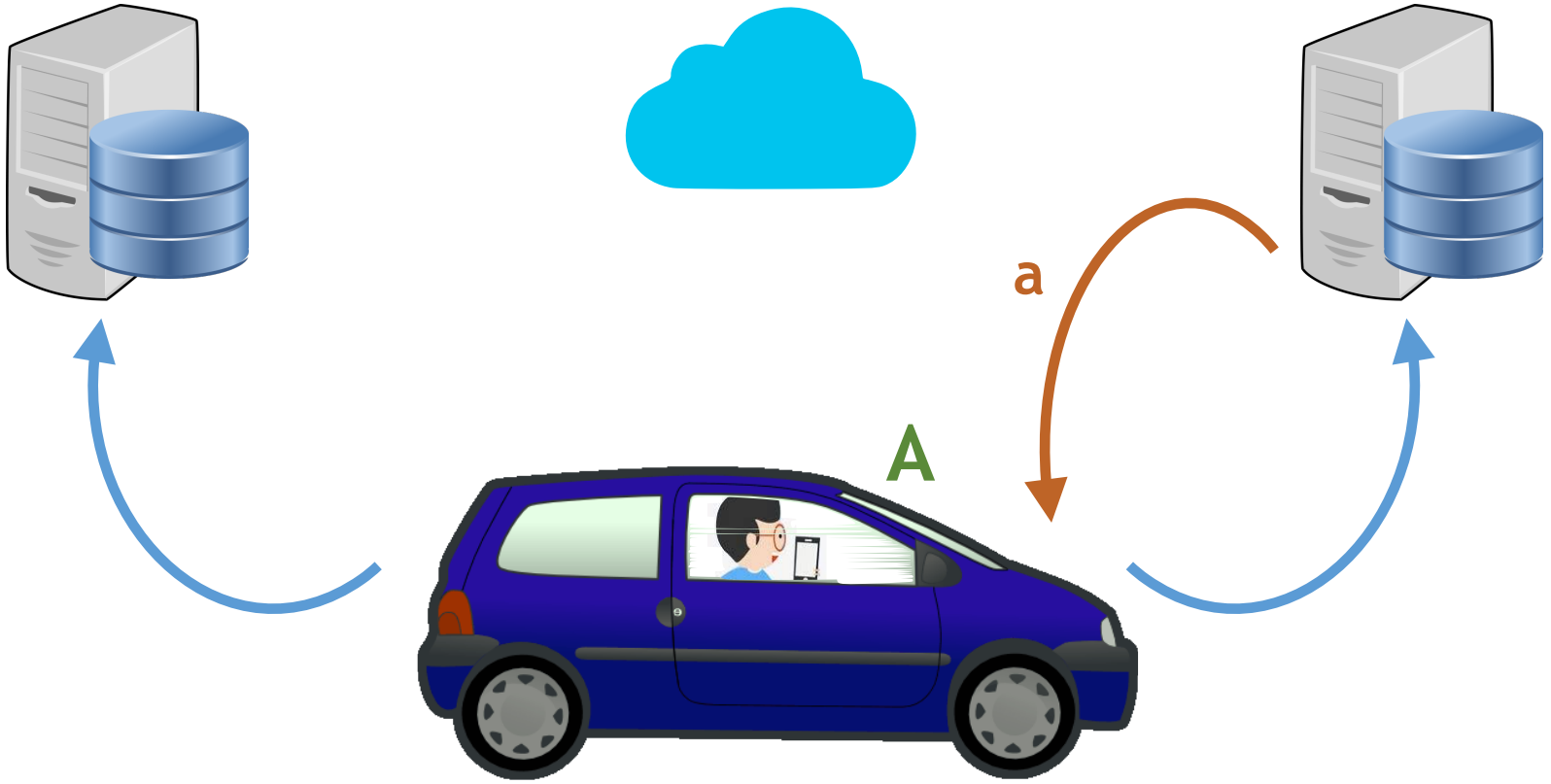
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



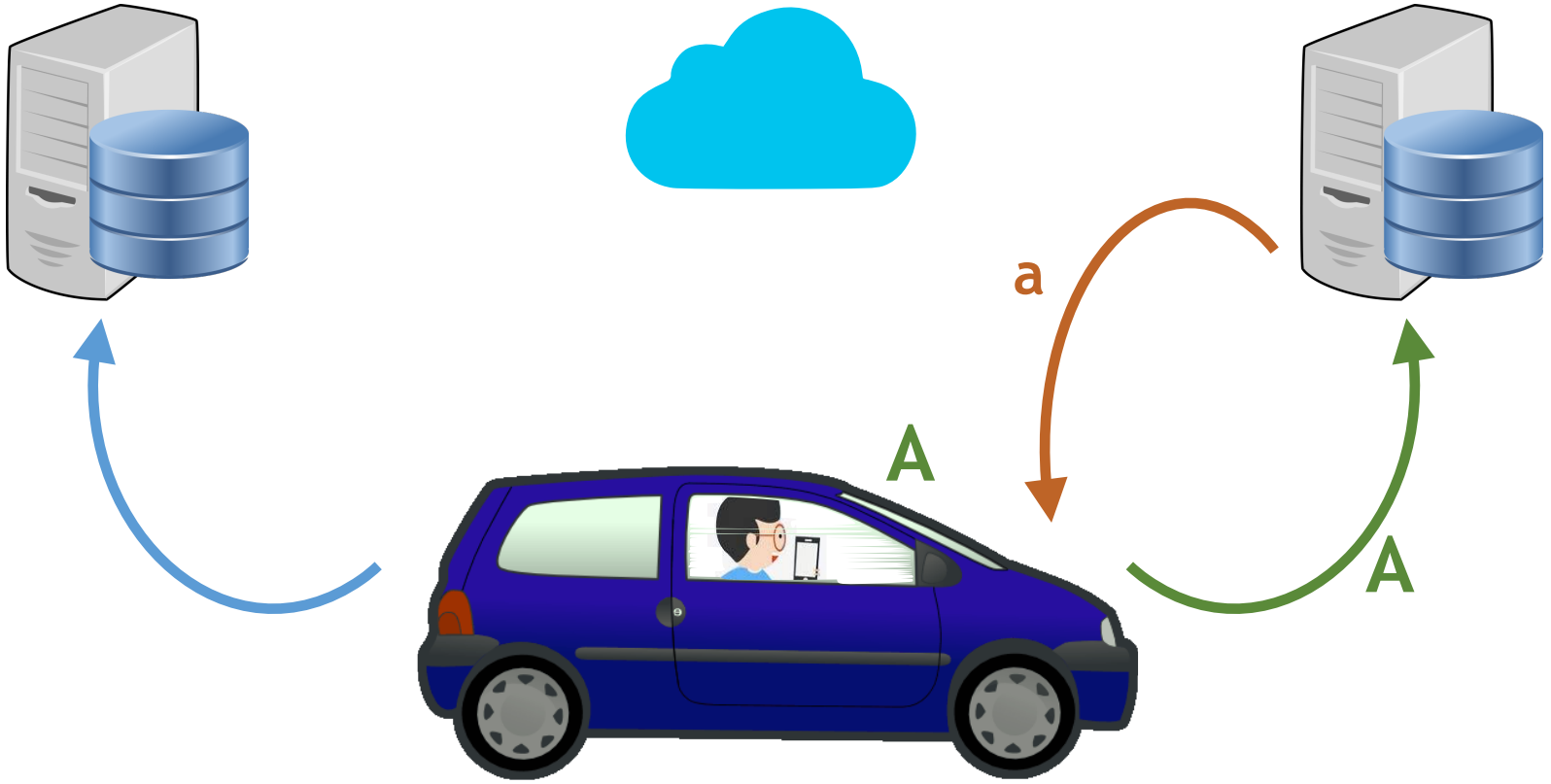
External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



External Sources

- Determinism is only useful if replicas receive the same sequence of inputs



3rd party services can produce non-deterministic results and still work with Shadow!

Determinism Summary

Non-Determinisim	Solution	Application Limitations
Multi-threading	Weak determinism	No data races
OS Events	Deterministic syscalls	Time skew No IPC or kernel devices
Client Inputs	Use client as primary	None
External Sources	Use client as primary	None

Outline

- Motivation
- Determinism to the rescue
- Shadow: Deterministic containers
- Evaluation

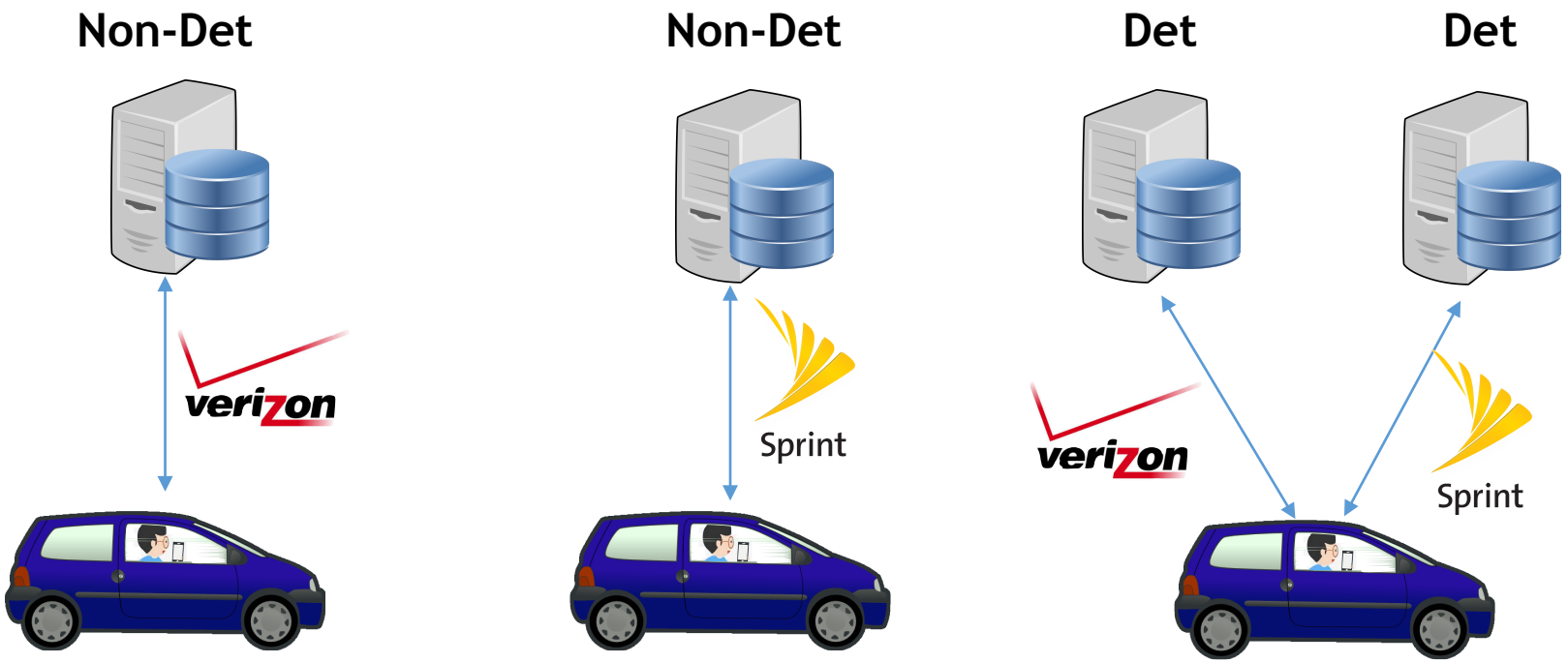
Evaluation

- Test Application
  - Vehicle Caravan



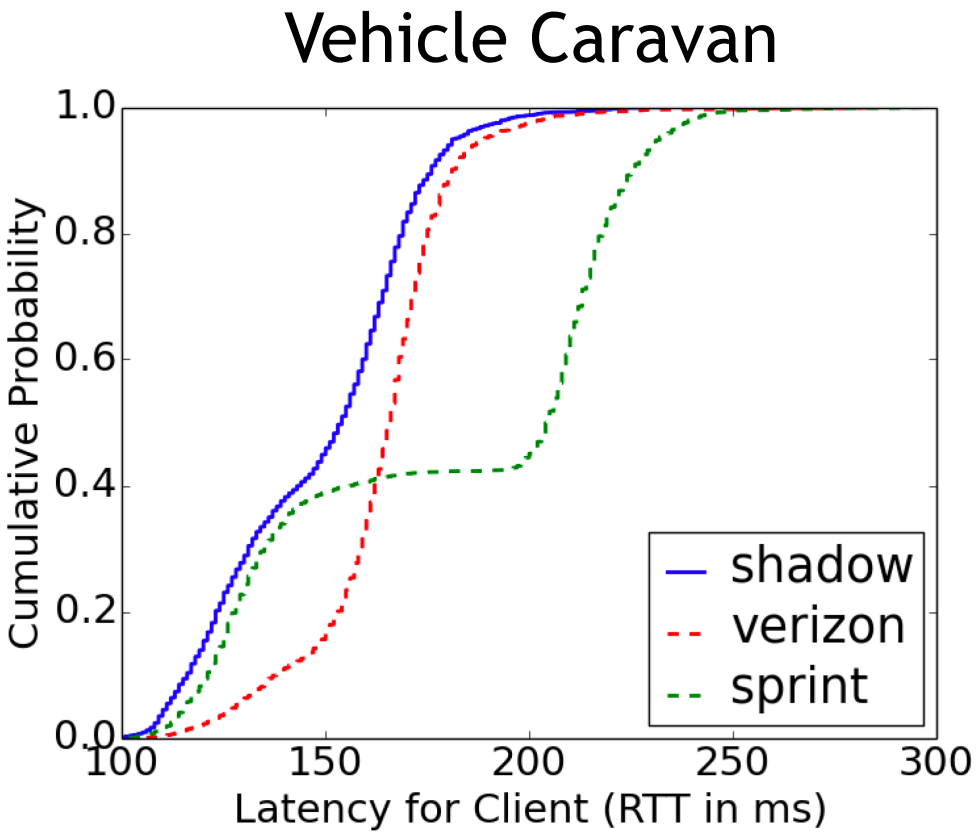
Jitter Results

- Emulate suburban network traces from VNperf[1]
- Evaluate quality of experience for the client using response time from server

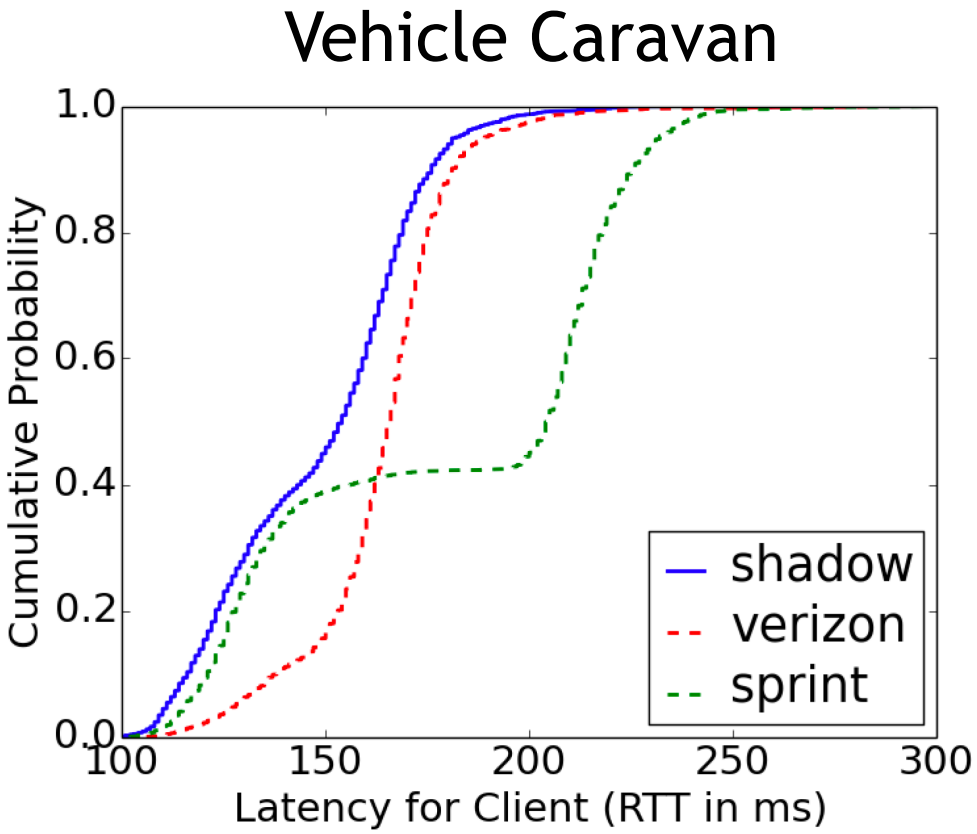


[1] Lee, H., Flinn, J., And Tonshal, B. Raven: Improving interactive latency for the connected car.

Jitter Results



Jitter Results



Shadow reduces the median response by up to 25 percent

Discussion

- Feedback
  - Are Shadow's limitations reasonable to place on applications?
  - How prevalent does the community think single-client applications will be in edge deployments?
  - What other applications could benefit from a deterministic environment?
- Controversial points
  - Applications must fit into Shadow's restrictions (e.g. no IPC, clock skew, race-free)
- Everything falls apart…
  - Performance overhead of determinism outweighs the benefits
  - If the majority of applications cannot be modified to fit into Shadow's restrictions