

Reconfiguration of replicated systems



Technische
Universität
Braunschweig

Leander Jehl

Reconfiguration

Part 1 - General approaches

- Replicated systems
- What is Reconfiguration
- How to perform a reconfiguration (1)
- How to specify a reconfiguration
- Concurrent reconfigurations
 - Totally ordered configurations
 - CRDT based reconfigurations

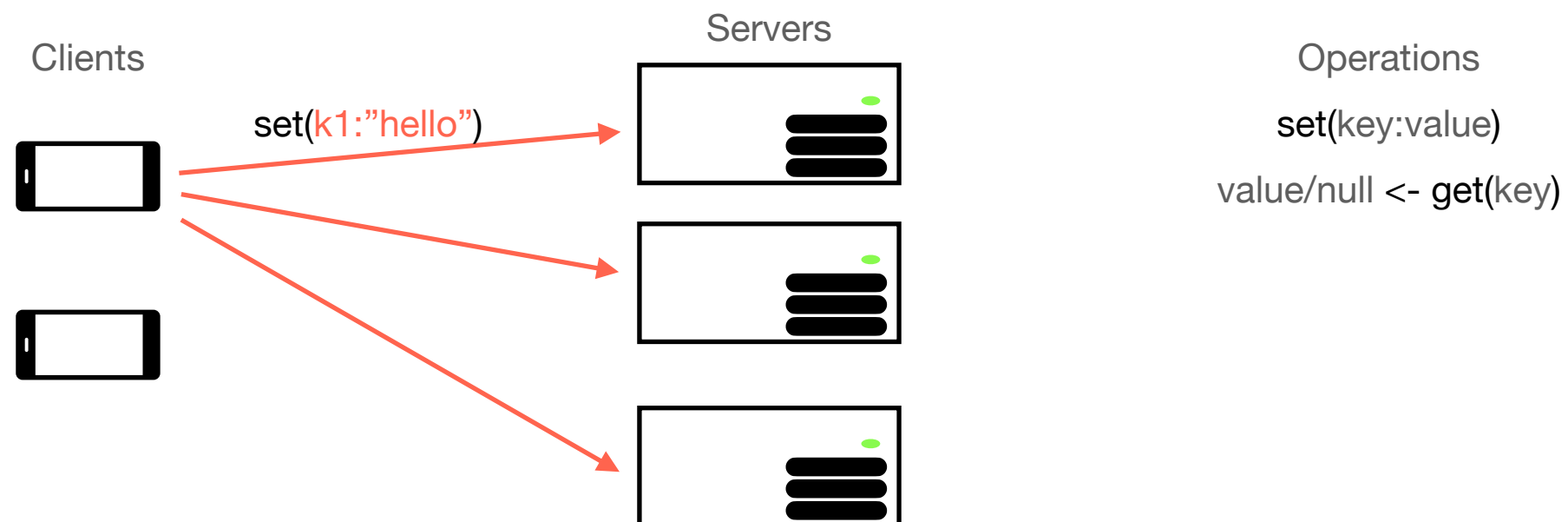
Reconfiguration

Part 2 - Special cases

- Reconfiguration in sharded systems
- Reconfiguration in BFT
 - Directional
 - Trusted

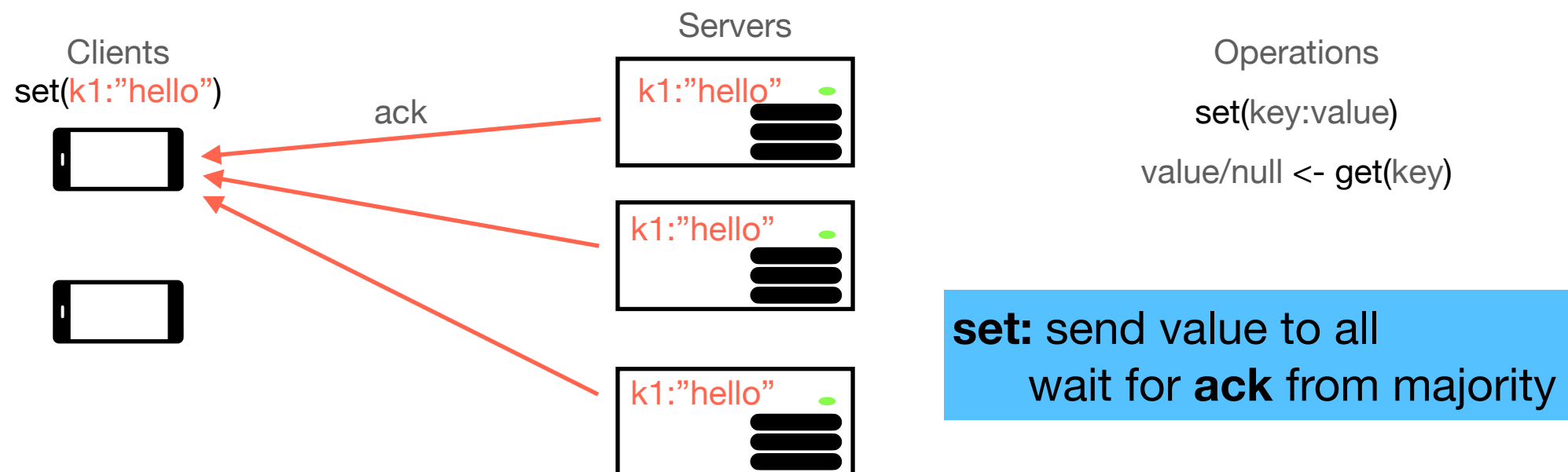
Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



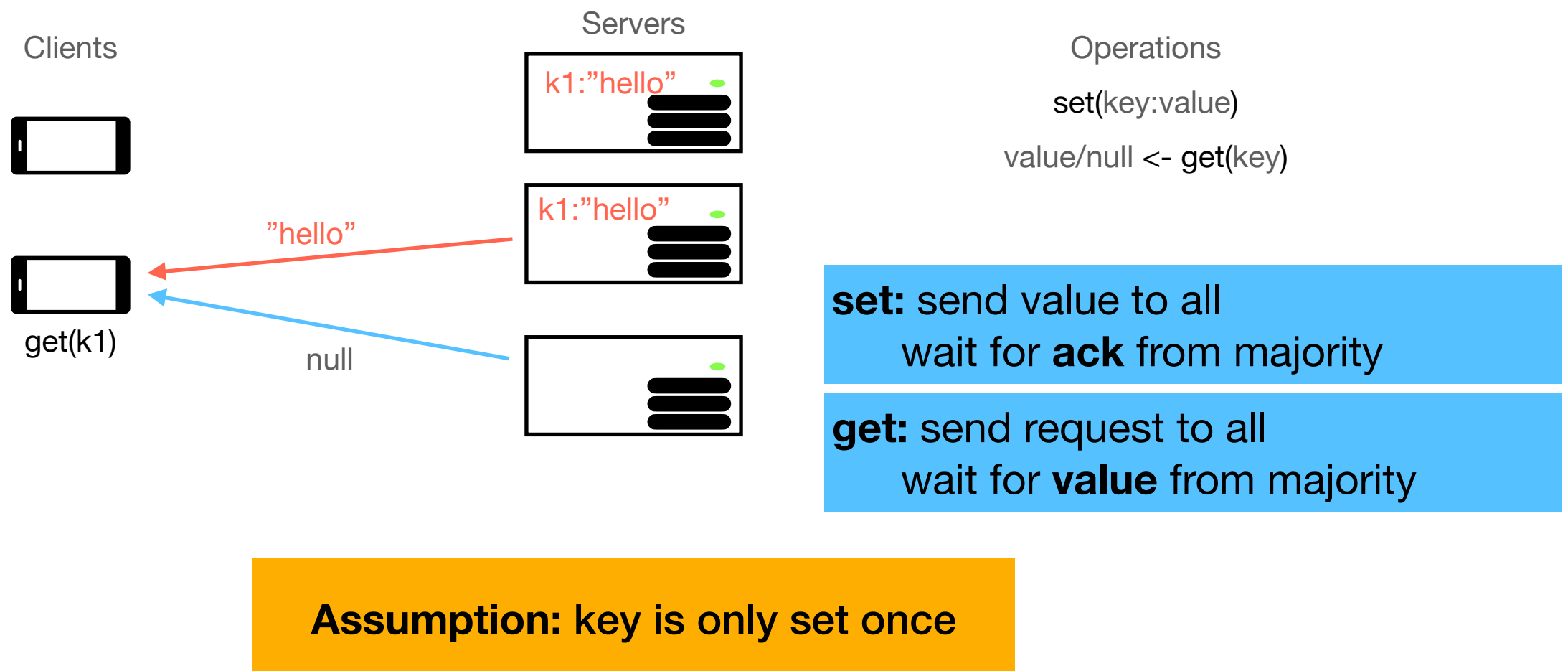
Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



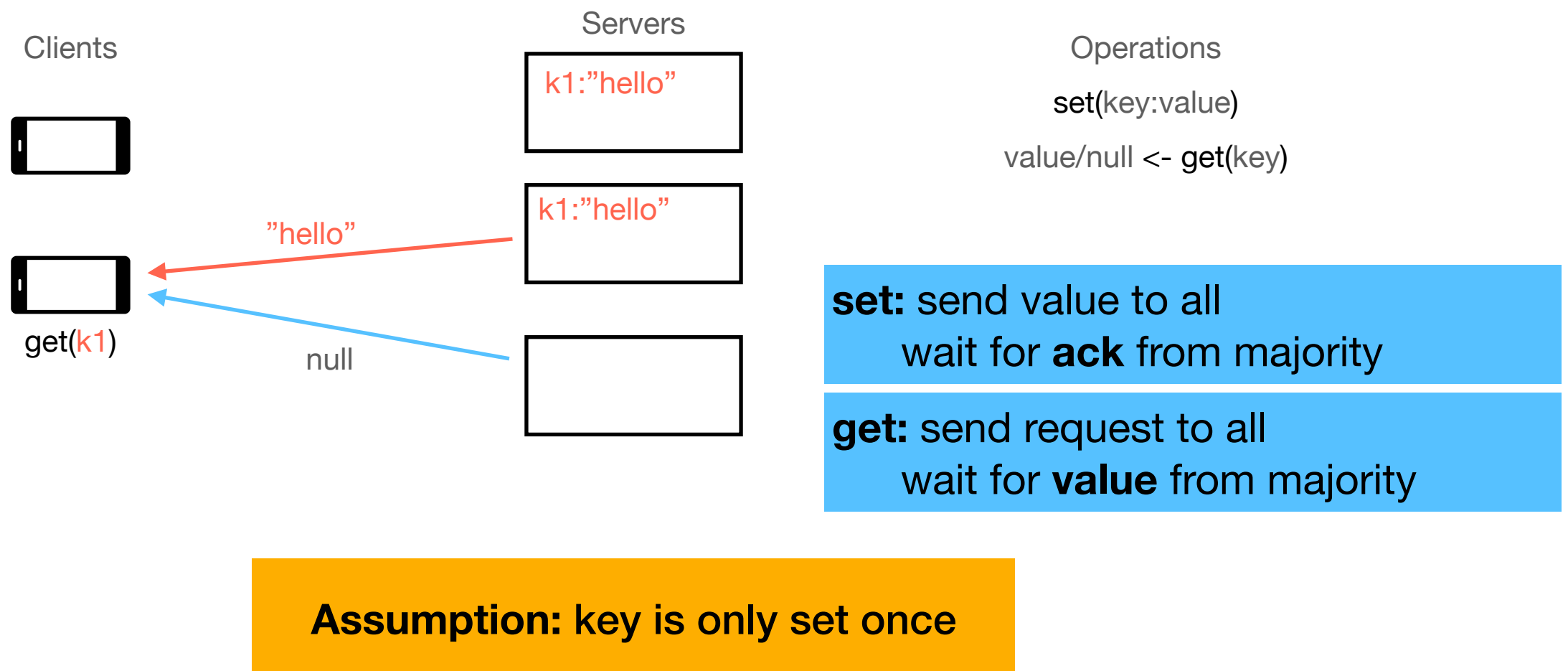
Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



Replicated system

- Nodes store the same state for fault tolerance and scalability
Storage systems, Replicated state machines, Blockchains
- Running example: Majority quorum key value store.



Reconfiguration

Reconfiguration is a change in the set of servers of a system or in the way these servers are organised.

Servers

k1:"hello"

k1:"hello"

New

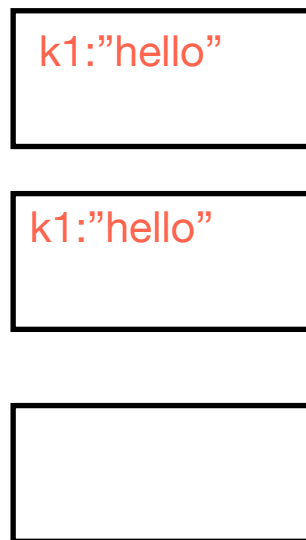
set: send value to all
wait for **ack** from majority

get: send request to all
wait for **value** from majority

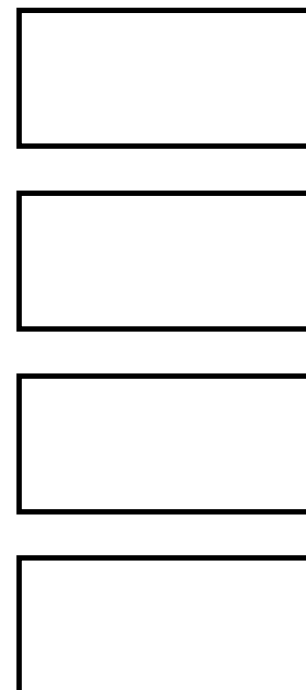
Reconfiguration

Reconfiguration is a change in the set of servers of a system or in the way these servers are organised.

Servers 1



Servers 2



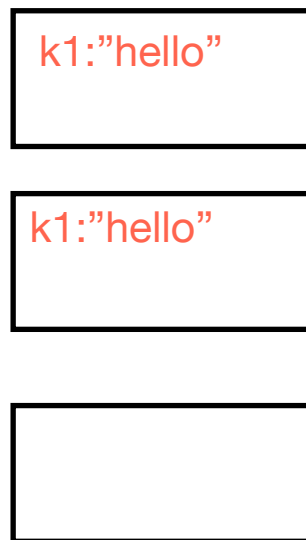
set: send value to all
wait for **ack** from majority

get: send request to all
wait for **value** from majority

Reconfiguration

Reconfiguration is a change in the set of servers of a system or in the way these servers are organised.

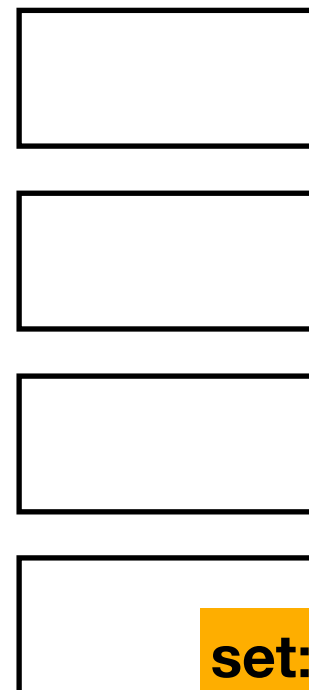
Configuration C1



set: send value to all
wait for **ack** from majority

get: send request to all
wait for **value** from majority

Configuration C2



set: send value to all
wait for **ack** from **3**

get: send request to all
wait for **value** from **2**

Reconfiguration

Reconfiguration is a change in the set of servers of a system or in the way these servers are organised.

- Adding nodes
- Removing nodes
- Replace nodes
- Change read/write configuration (quorums)

Reconfiguration

Where to store configurations?

1. Servers:

- Each servers knows which configuration it is in.
- Servers may maintain separate state in different configurations.

Abstraction: Each server is part of only one configuration!

- Each request includes, which reconfiguration it is sent in.

But: Clients need to know configuration, to count quorum.

Reconfiguration

Where to store configurations?

1. Clients:

- Servers reply to any request, do not care about configuration.
- But Clients need to learn about configurations.

Idea: Distribute configurations by writing to a specific key.

May introduce special semantics for that key.

How to do a Reconfiguration

- Stop the old configuration

After complete reconfiguration, no `get` or `set` should be performed on old configuration

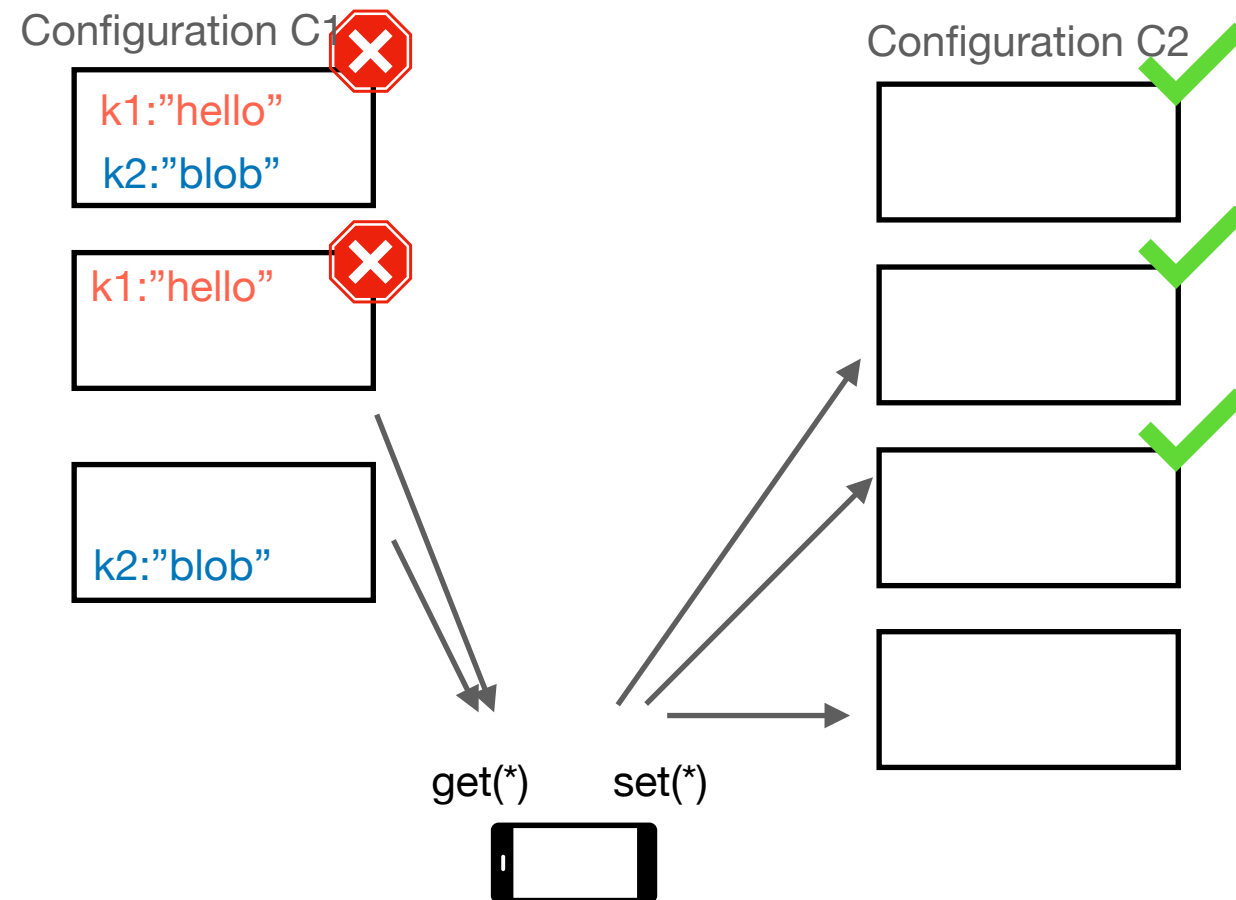
- Transfer state

`get` from old and `set` to new

- Start new configuration

How to do a Reconfiguration

- Stop the old configuration
- Transfer state
- Start new configuration



How to do a Reconfiguration

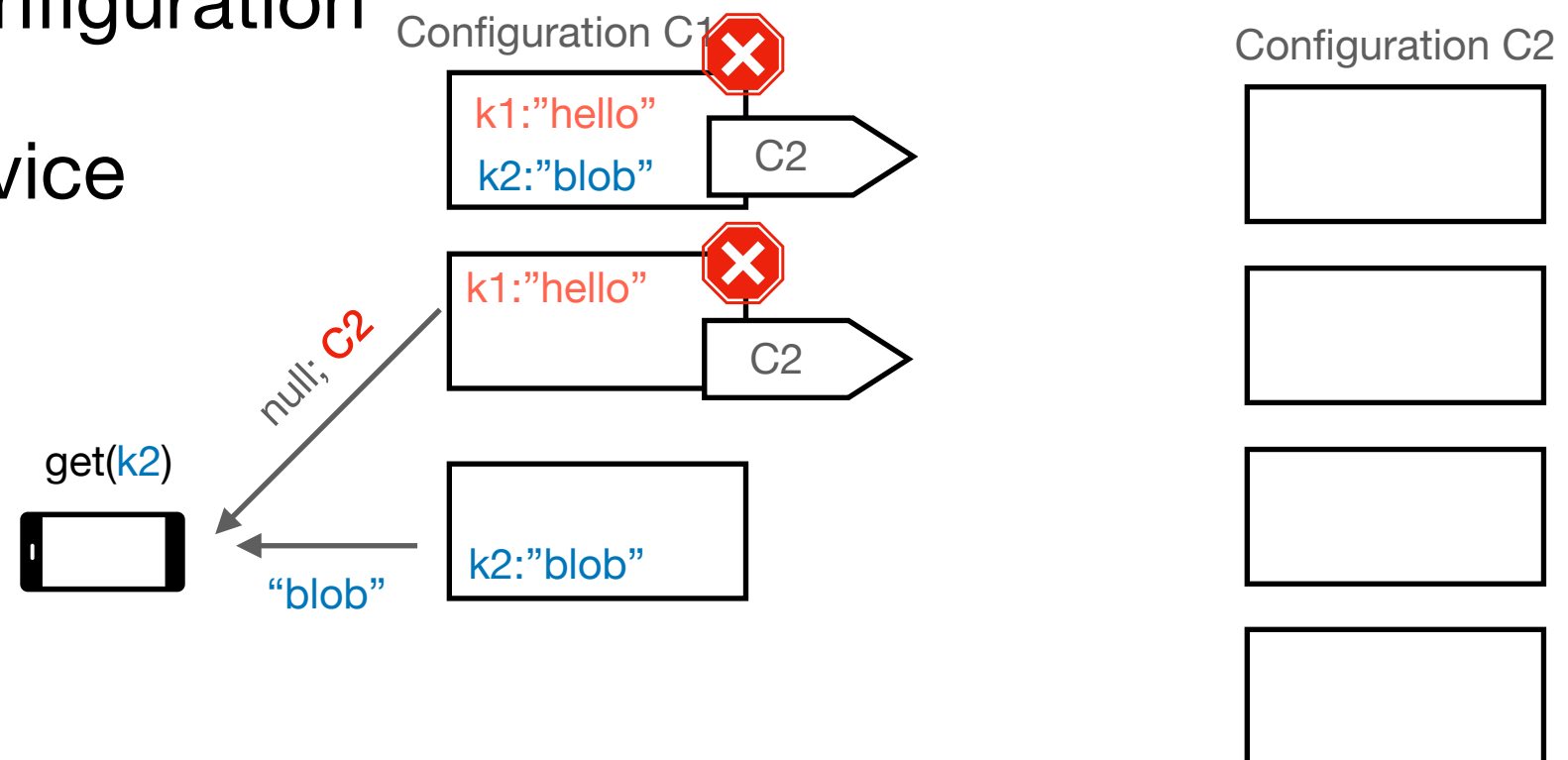
Problems

- How to find the new configuration?
- How to start all new servers?
- How to not stop `get` and `set`?
- How to optimize state transfer?
- How to deal with multiple new configurations?

How to do a Reconfiguration

Problems

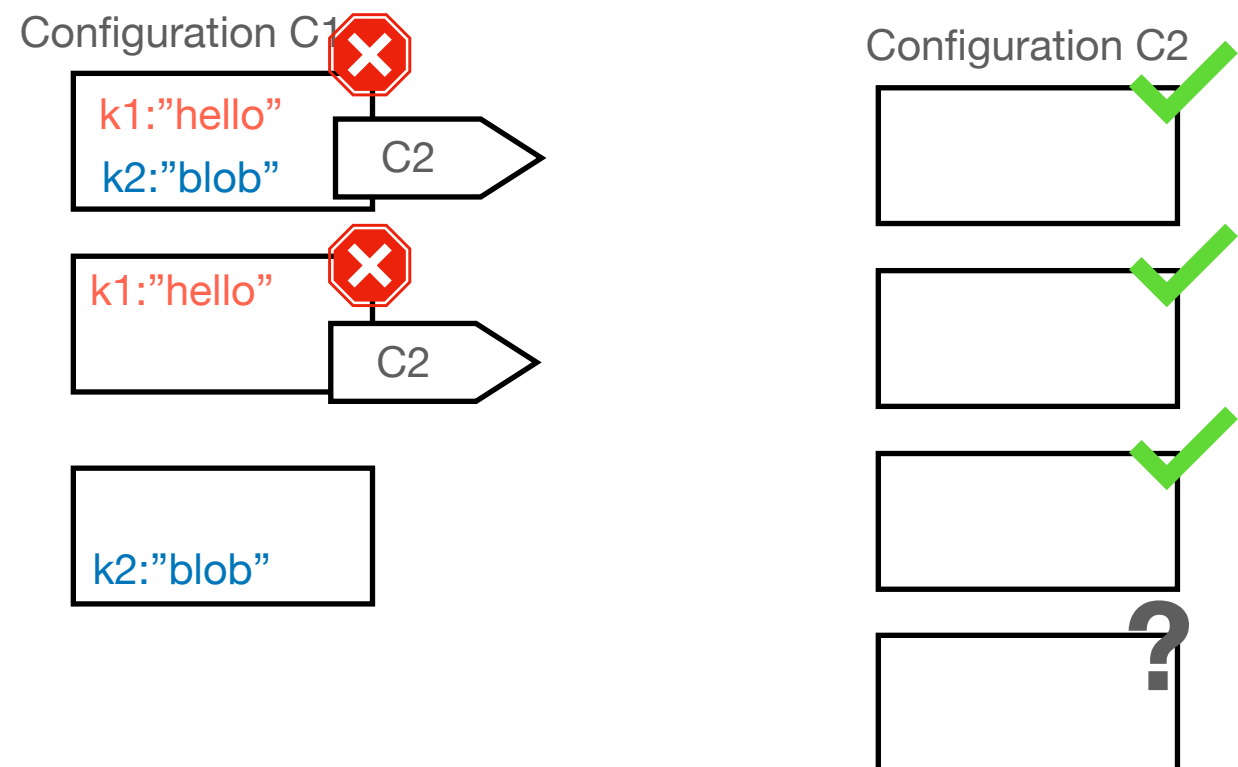
- How to find the new configuration?
- tell the old configuration
- discovery service



How to do a Reconfiguration

Problems

- How to start all new servers?

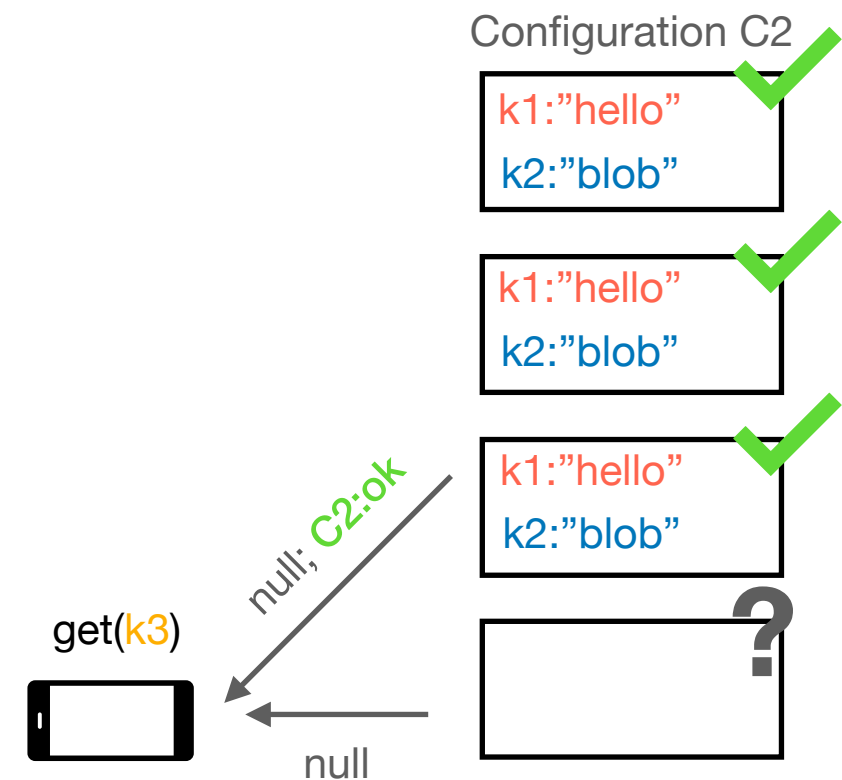


How to do a Reconfiguration

Problems

- How to start all new servers?
- Can answer clients before started.

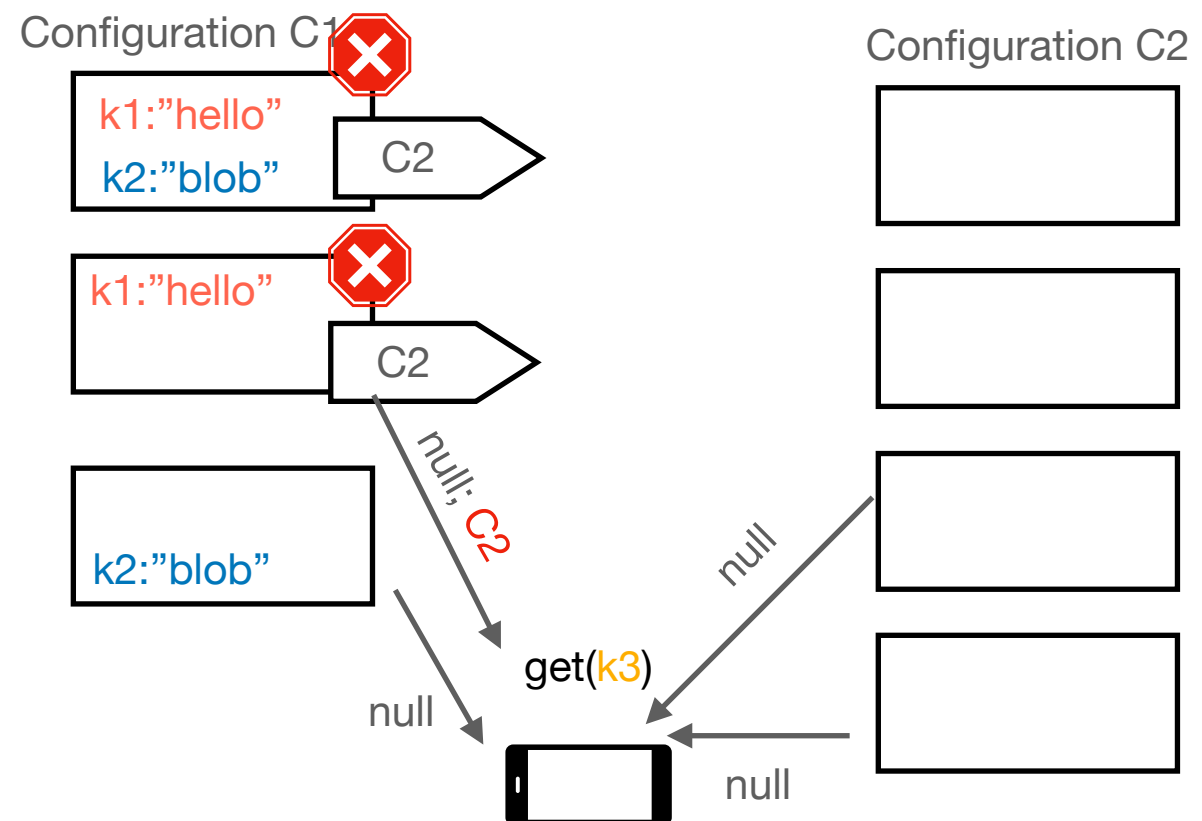
May result in polling



How to do a Reconfiguration

Problems

- How to not stop get and set?
 - Contact old and new configuration.
- Some servers may be in old and new



How to do a Reconfiguration

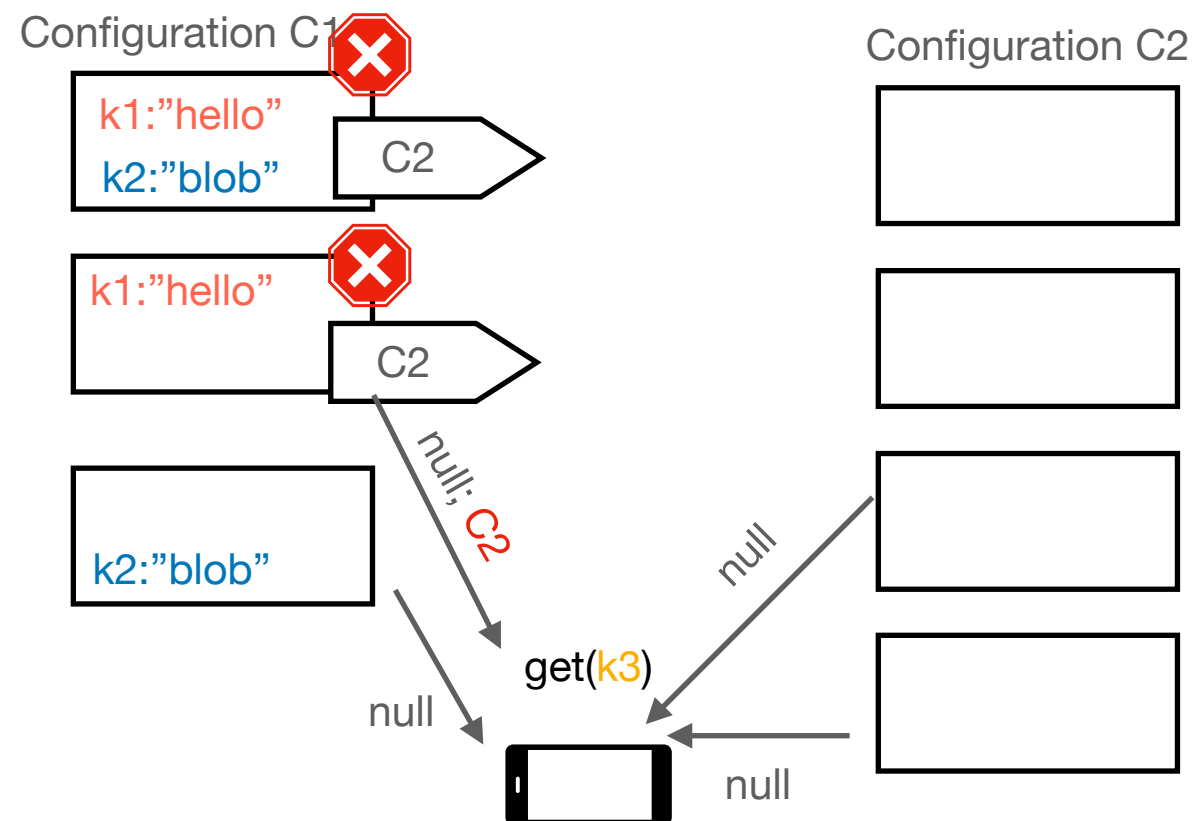
Problems

- How to not stop get and set?
 - Contact old and new configuration.
- Some servers may be in old and new

Optimization for:

- Adding/removing one server
- Majority quorums

Contact old or new config.



How to do a Reconfiguration

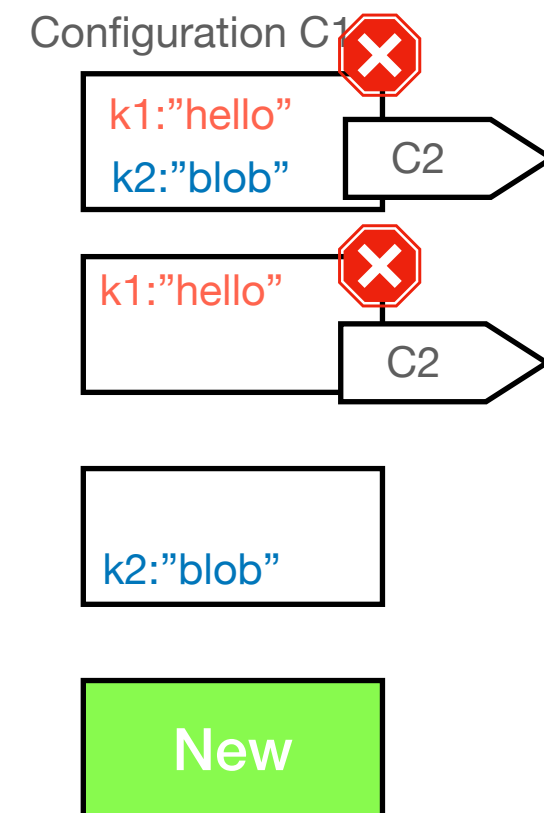
Problems

- How to not stop get and set?
 - Contact old and new configuration.
- Some servers may be in old and new

Optimization for:

- Adding/removing one server
- Majority quorums

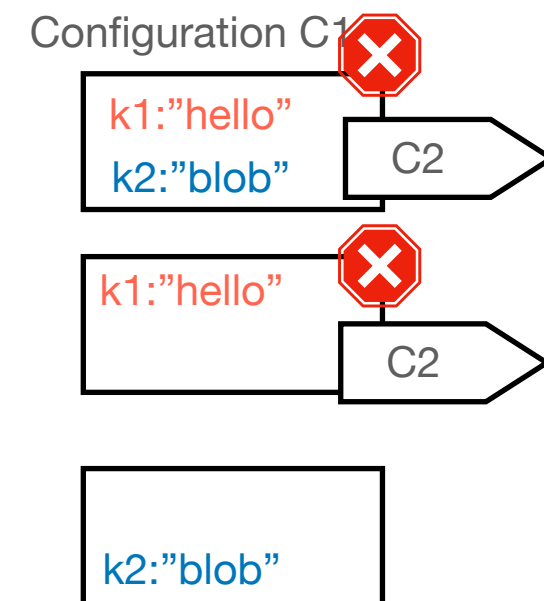
Contact old or new config.



How to do a Reconfiguration

Problems

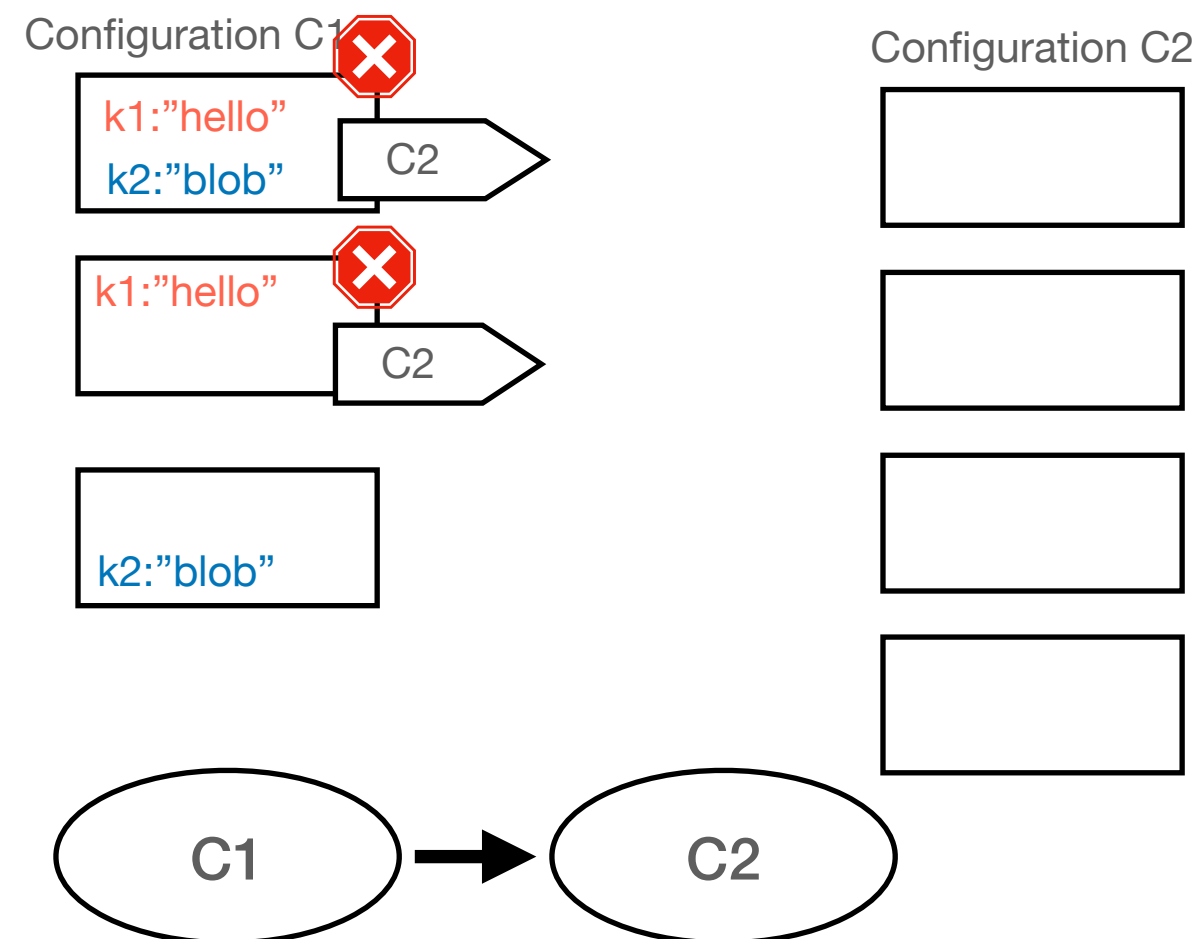
- How to optimize state transfer
 - Shedule reconfiguration in advance
- Initialize new configuration and transfer old state
- On reconfiguration, transfer Δ



How to do a Reconfiguration

Problems

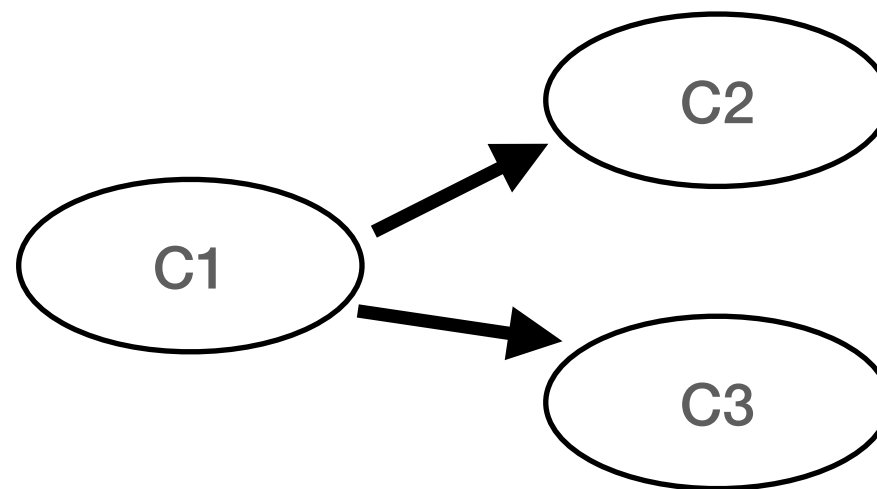
- How to deal with multiple new configurations?



How to do a Reconfiguration

Problems

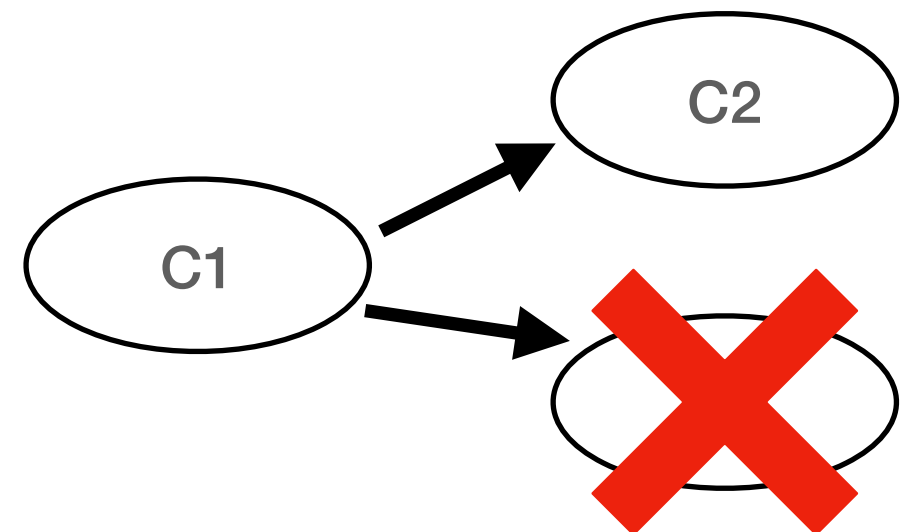
- How to deal with multiple new configurations?



How to do a Reconfiguration

Problems

- How to deal with multiple new configurations?
 - Prevent through:
 - single (unfailable) sys-admin
 - each configuration choose its successor (consensus)



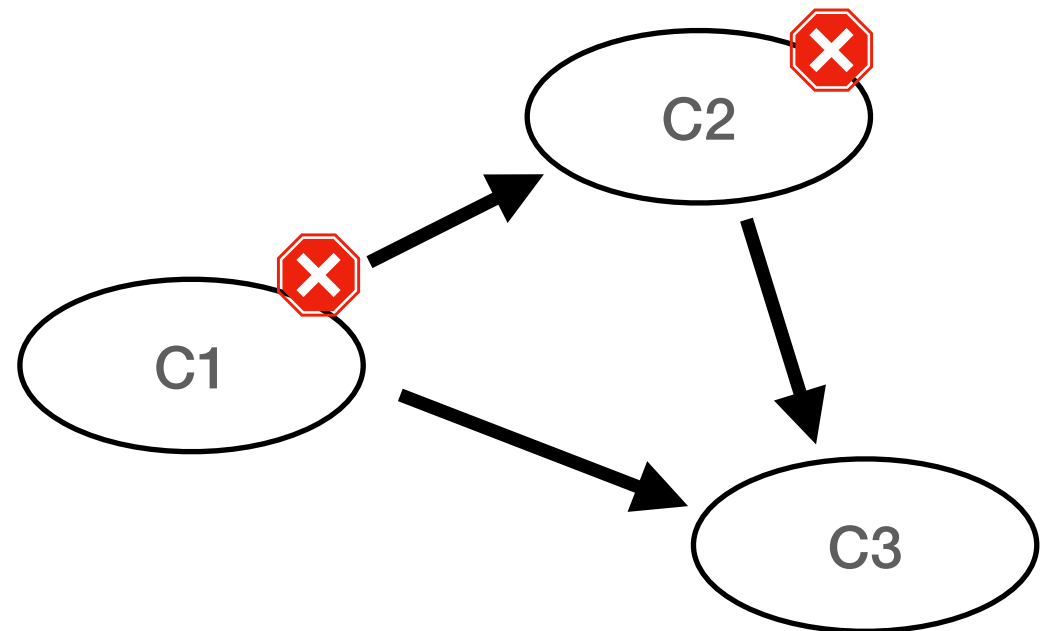
How to do a Reconfiguration

Problems

- How to deal with multiple new configurations?

Reconfiguration

- Stop the old config and mark next
- Transfer state
get t from old and set to new
- Start new configuration



Can configure from multiple to one new configuration.

But we need a priority.

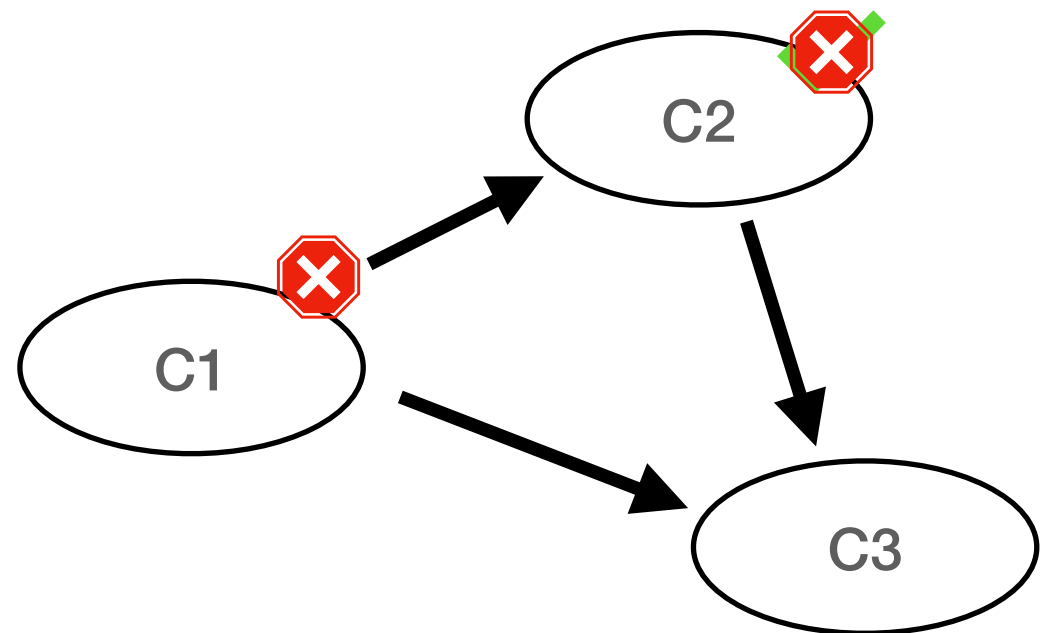
How to do a Reconfiguration

Problems

- How to deal with multiple new configurations?

Reconfiguration

- For all old configs:
 - Stop the old config and mark next
 - Add lower priority configs to old
 - On higher priority config abort
- Transfer state
*get t from **all** old and and set to new*
- Start new configuration



How to do a Reconfiguration

Problems

- How to deal with multiple new configurations?

Reconfiguration

- For all old configs:
 - Stop the old config and mark next
 - Add lower priority configs to old
 - On higher priority config abort
- Transfer state
*get t from **all** old and and set to new*
- Start new configuration

Priority

- e.g. Timestamps or sequence numbers
- Process IDs for tie breaking

Totally ordered configurations

How to specify a reconfiguration

1. Specify configuration:

$\langle \text{servers:} \{s_1, s_2, \dots\}, \text{quorums} : \dots, \text{priority} \rangle$

2. As CRDT: conflict resolution data types

e.g. 2P-SET: $\langle \text{added:} \{s_1, s_2, s_3, \dots\}, \text{removed:} \{s_1, s_2, \dots\} \rangle$

- Two configs. can be combined, e.g. union added and removed

How to specify a reconfiguration

1. Specify configuration:

$\langle \text{servers:} \{s_1, s_2, \dots\}, \text{quorums} : \dots, \text{priority} \rangle$

- Easy to do complex functionality (quorums, ...)
- Only complete specified configurations

2. As CRDT: conflict resolution data types

e.g. 2P-SET: $\langle \text{added:} \{s_1, s_2, s_3, \dots\}, \text{removed:} \{s_1, s_2, \dots\} \rangle$

- Two configs. can be combined, e.g. union added and removed
- No reconfigurations lost due to priority

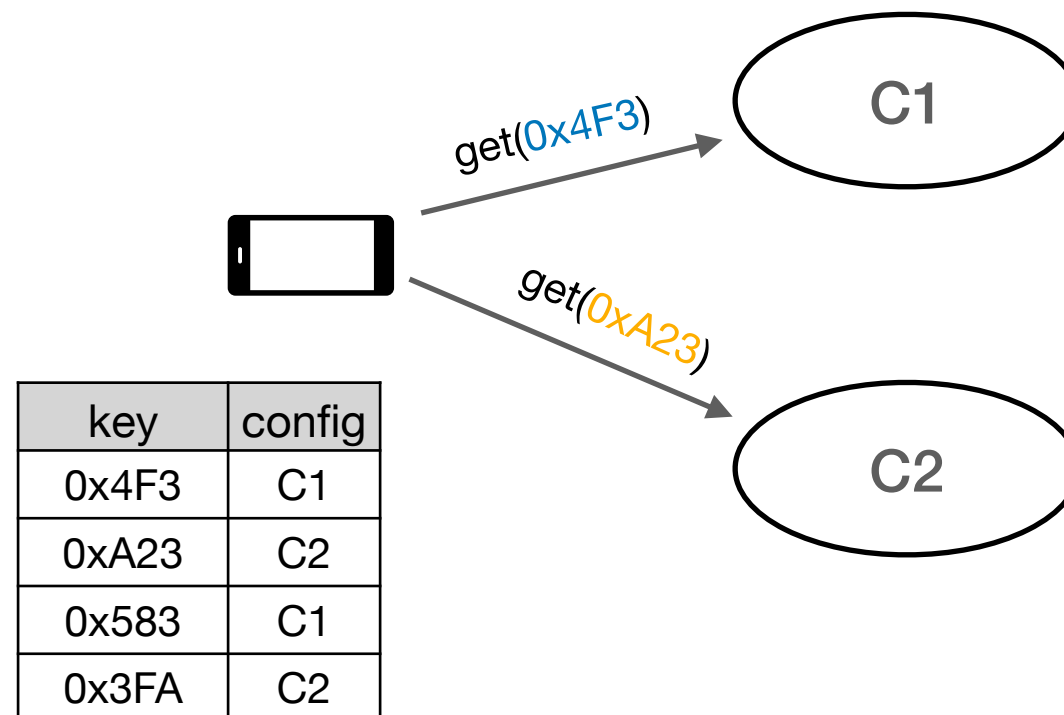
Reconfiguration

Part 2 - Special cases

- Reconfiguration in sharded systems
- Reconfiguration in BFT
 - For performance
 - For resilience

Sharded systems

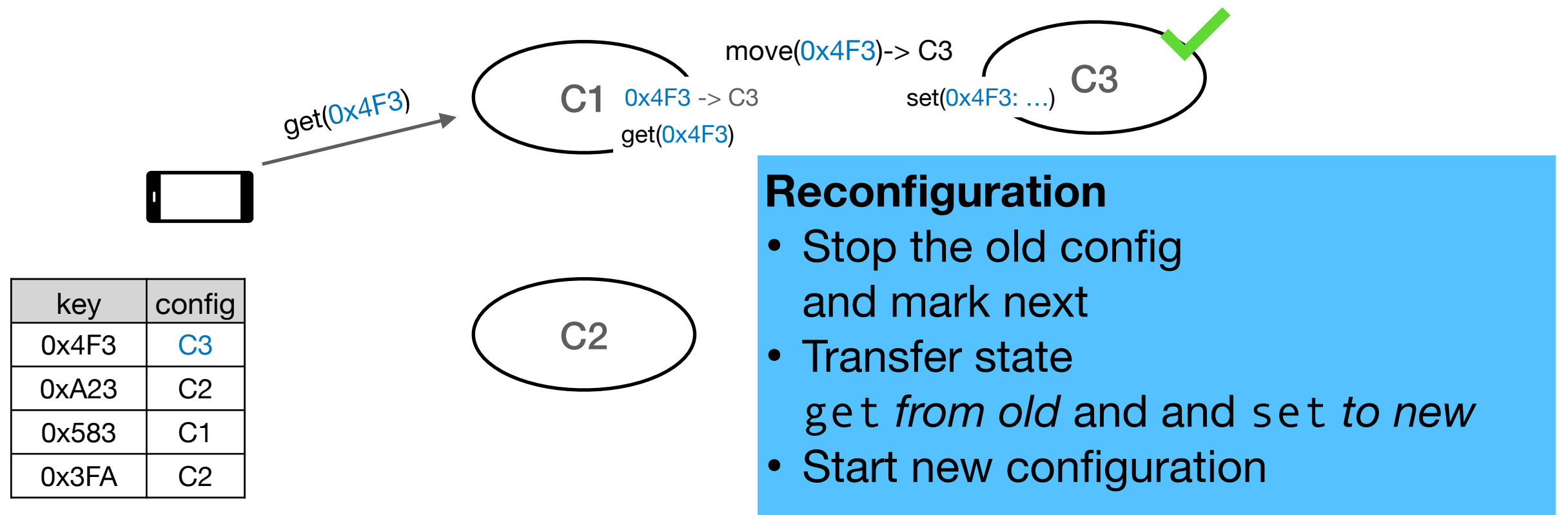
- Multiple configurations store different data:



Sharded systems

Reconfiguration

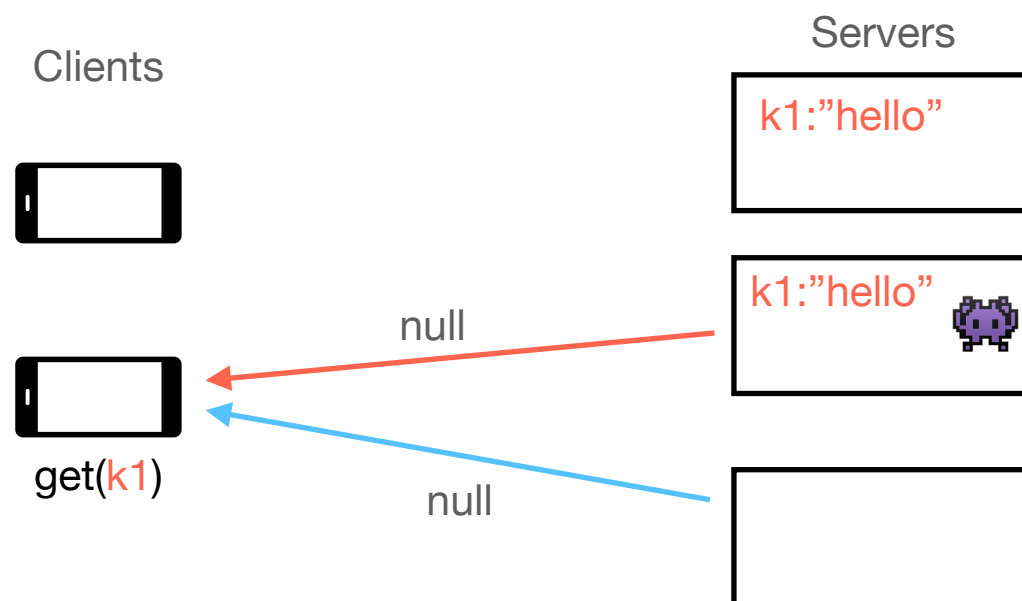
- Multiple configurations store different data:



- Reconfiguration: `move(key)` to new config

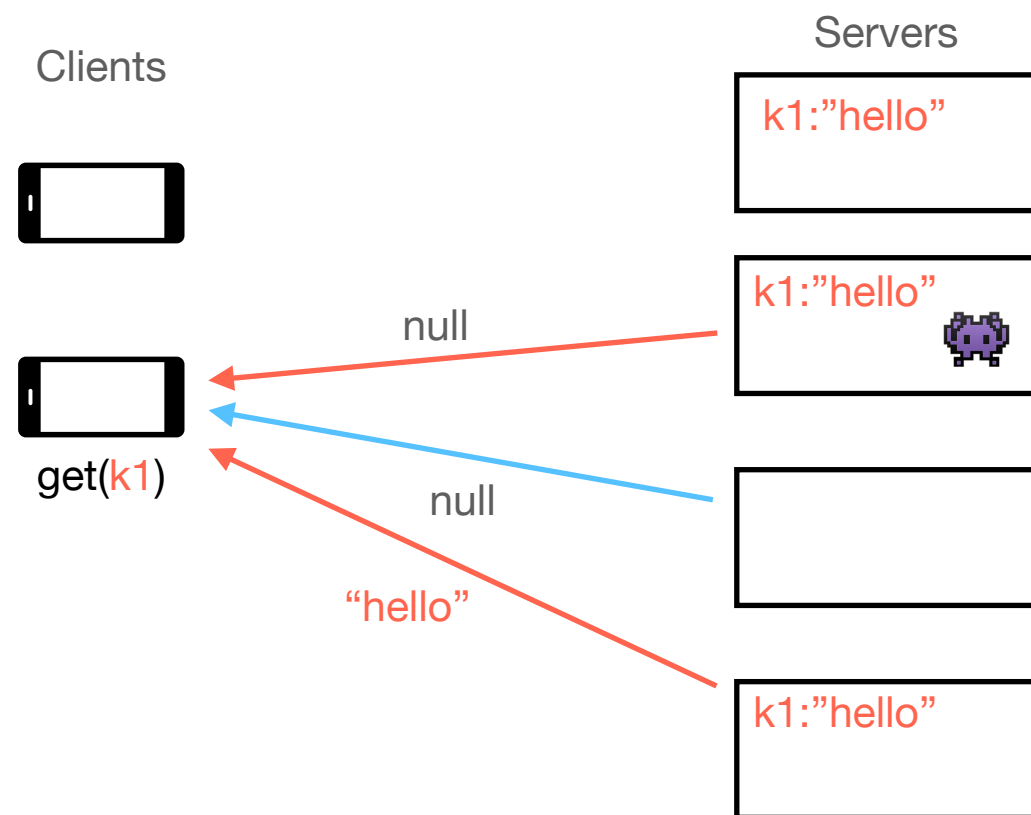
Reconfiguration in BFT

- In BFT systems servers may fail arbitrarily
may reply wrong



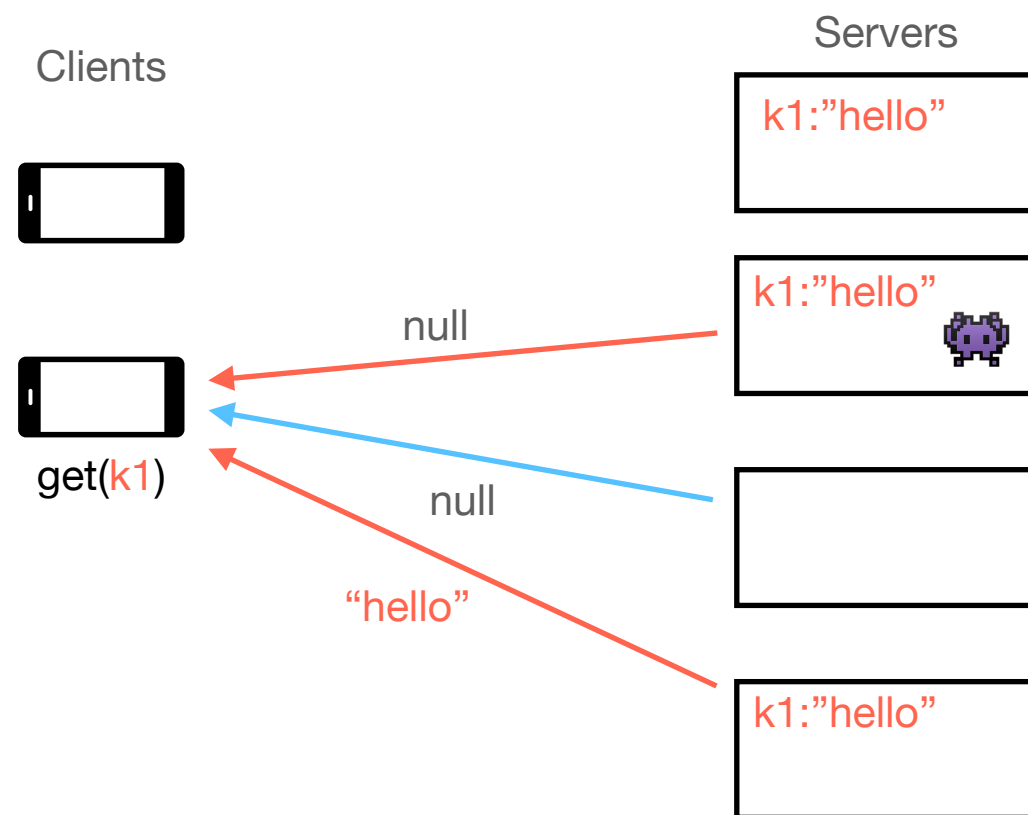
Reconfiguration in BFT

- In BFT systems servers may fail arbitrarily
may reply wrong



Reconfiguration in BFT

- In BFT systems servers may fail arbitrarily
may reply wrong



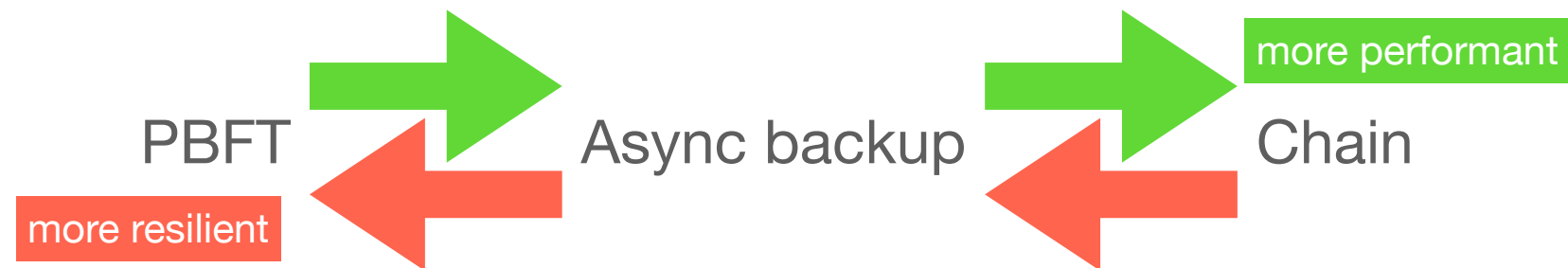
Reconfiguration

- Stop the old config and mark next
- Transfer state
get from old and set to new
- Start new configuration

Reconfiguration in BFT

When to configure?

- Resilience backup:



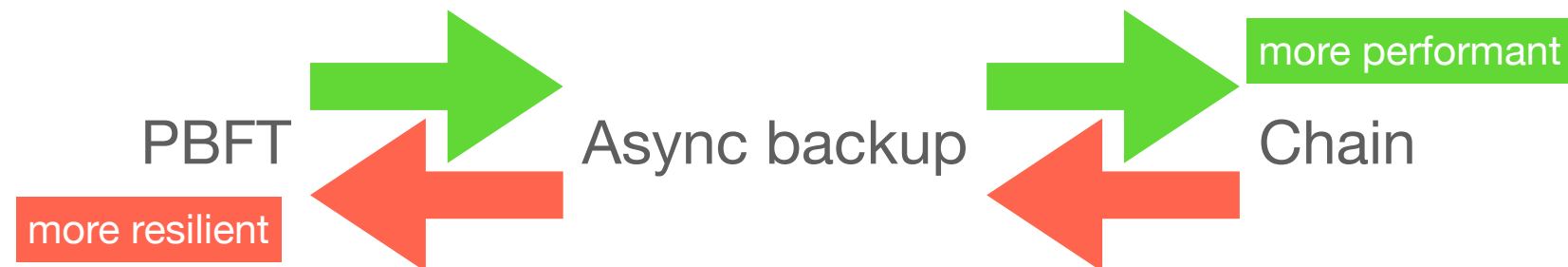
Anyone can trigger more resilience ←

Only periodically allow more performance →

Reconfiguration in BFT

When to configure?

- Resilience backup:



Anyone can trigger more resilience ←

Only periodically allow more performance →

- Trusted

Trusted subsystem decides on reconfiguration