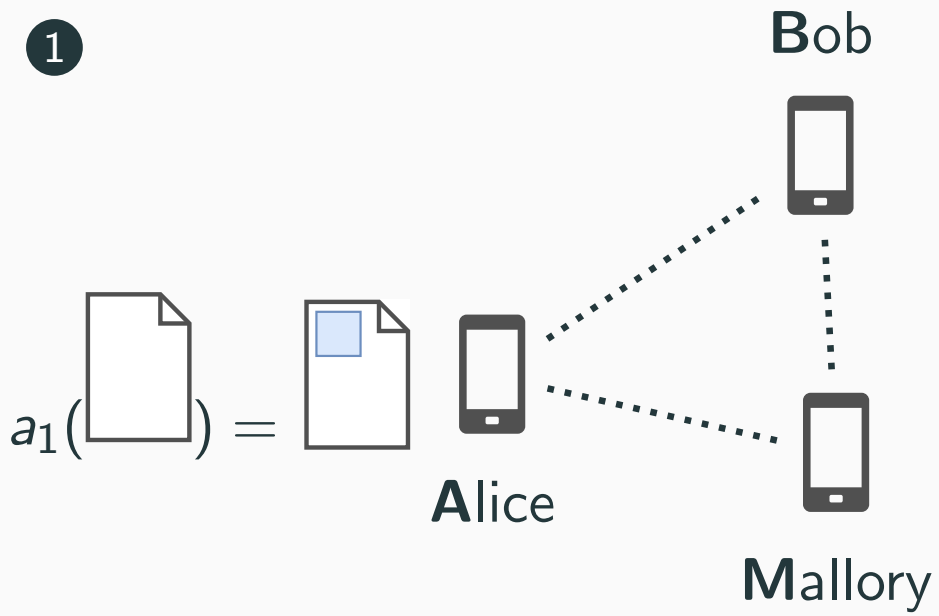
Prunable Authenticated Log and Authenticable Snapshot in Distributed Collaborative Systems

Victorien Elvinger

October 2018

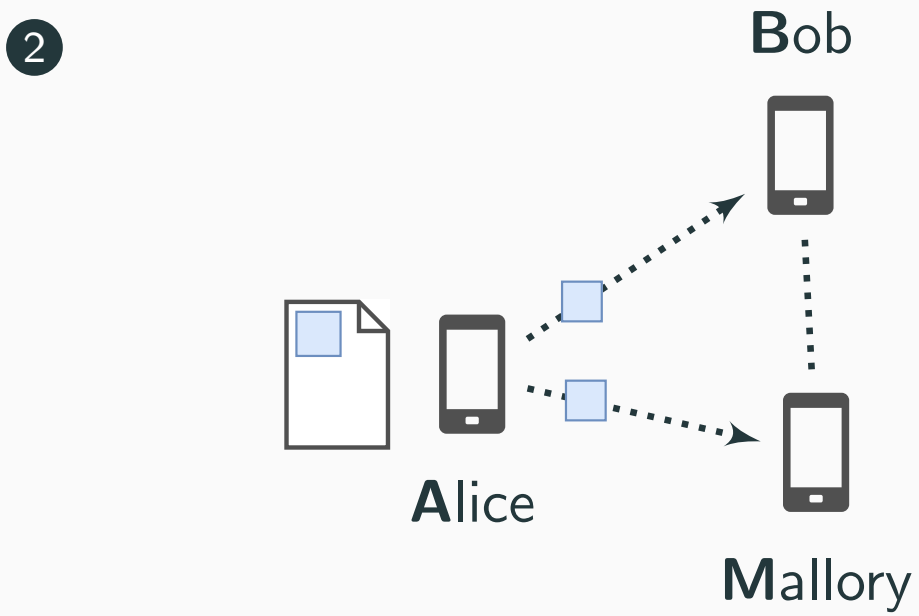

Supervised by Gérald Oster and François Charoy

Peer-to-Peer Collaborative Systems



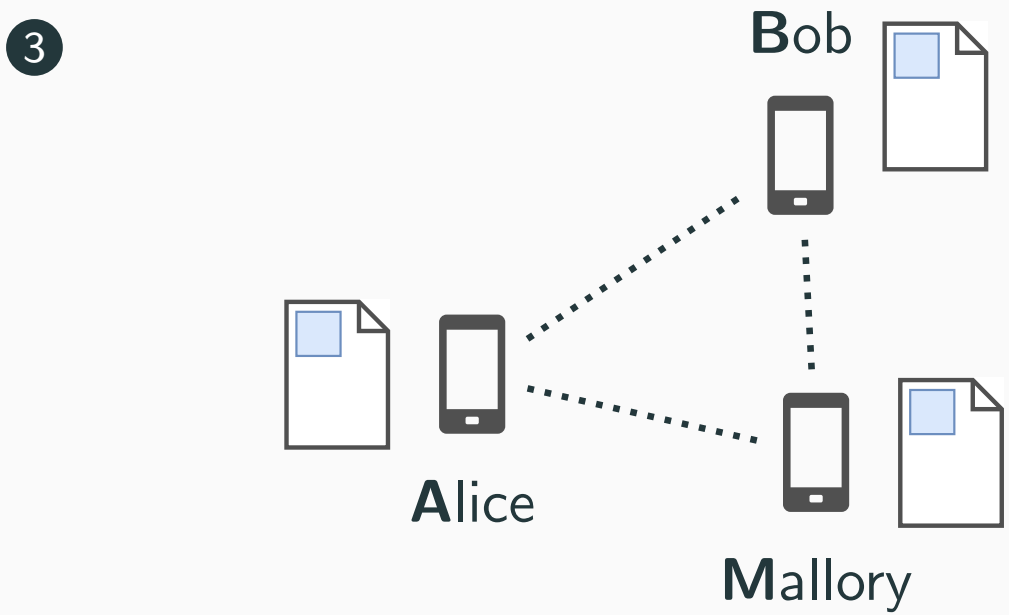
- participants issue **operations** to update a shared document

Peer-to-Peer Collaborative Systems



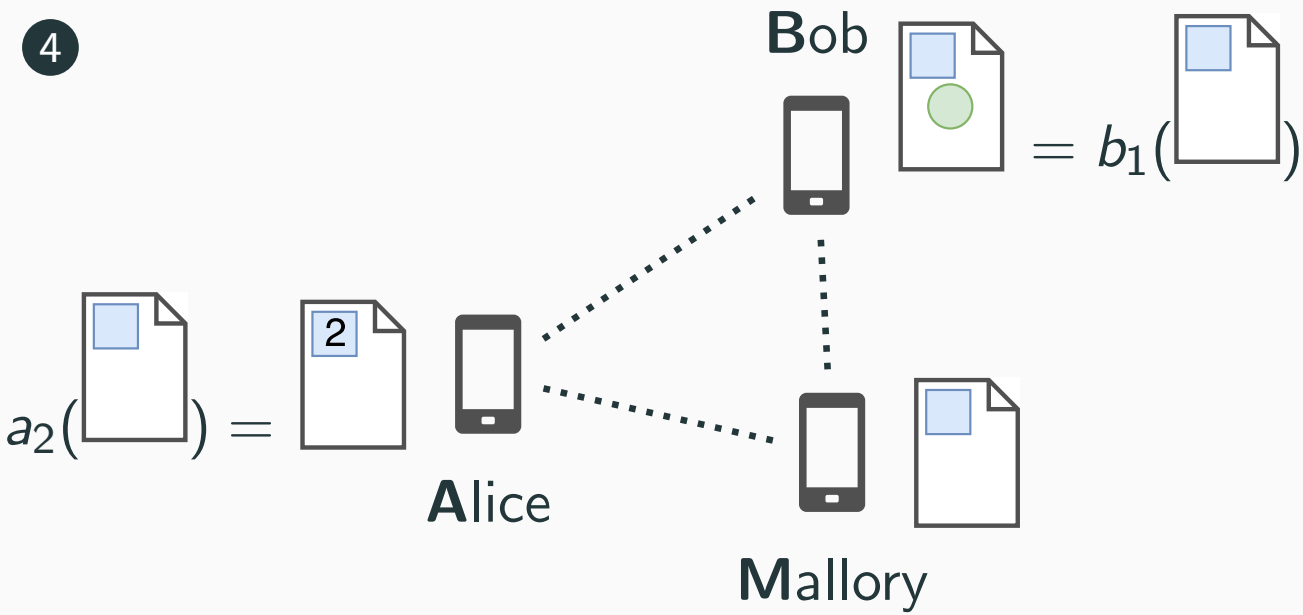
- participants issue **operations** to update a shared document

Peer-to-Peer Collaborative Systems



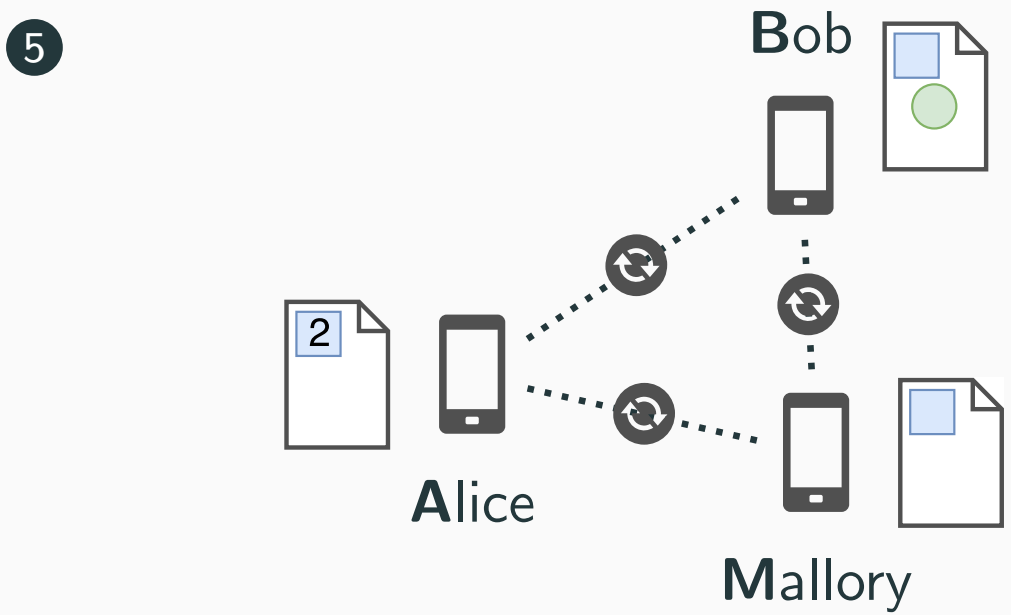
- participants issue **operations** to update a shared document
- participants have their **own copy** of **co-authored data**

Peer-to-Peer Collaborative Systems



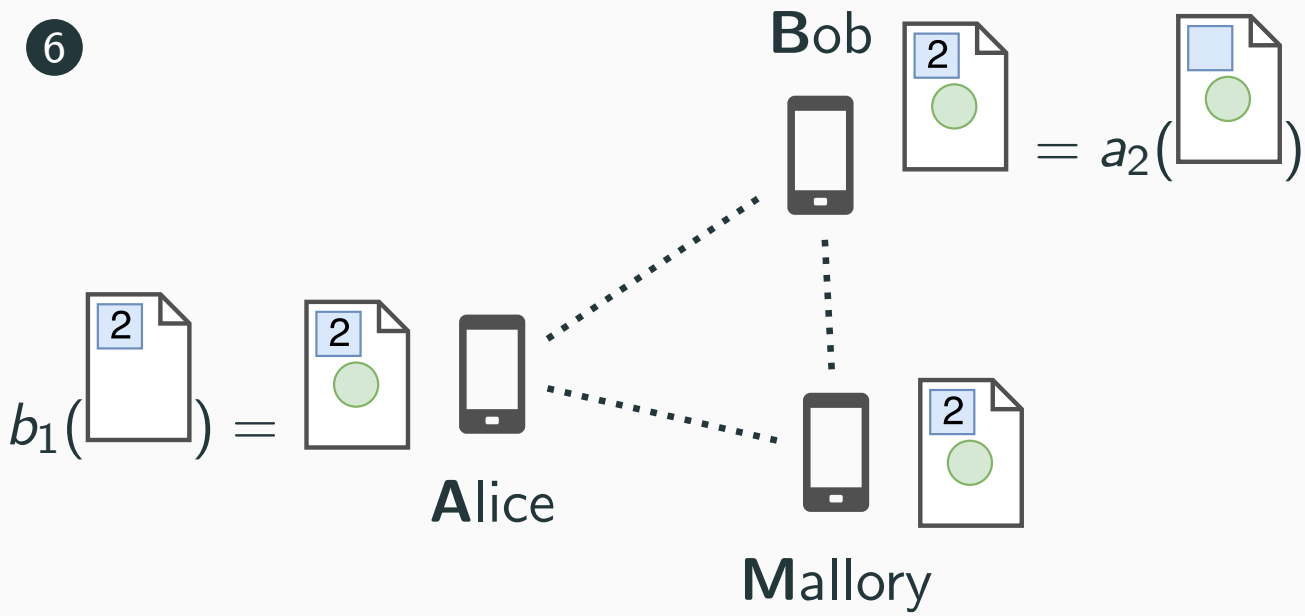
- participants issue **operations** to update a shared document
- participants have their **own copy** of **co-authored data**
  - **always-available**
  - continuously **diverge then converge** from others

Peer-to-Peer Collaborative Systems



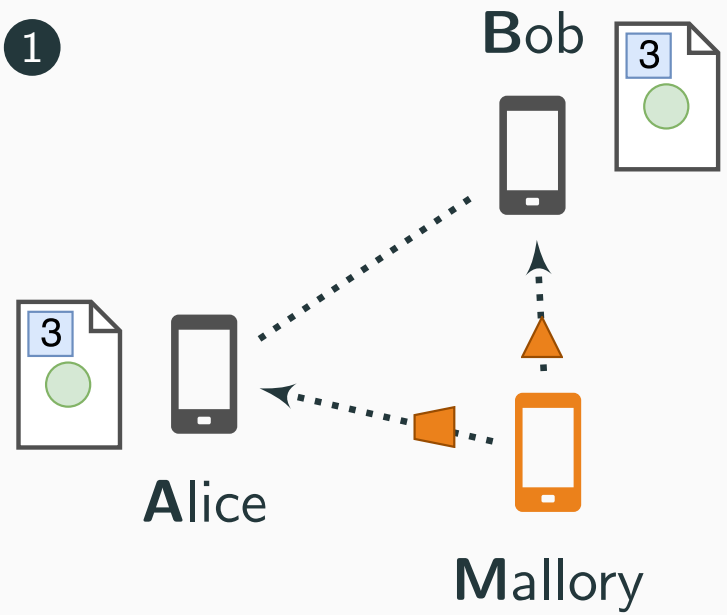
- participants issue **operations** to update a shared document
- participants have their **own copy** of **co-authored data**
  - **always-available**
  - continuously **diverge then converge** from others

Peer-to-Peer Collaborative Systems



- participants issue **operations** to update a shared document
- participants have their **own copy** of **co-authored data**
  - **always-available**
  - continuously **diverge then converge** from others
  - convergence as a **liveness property**

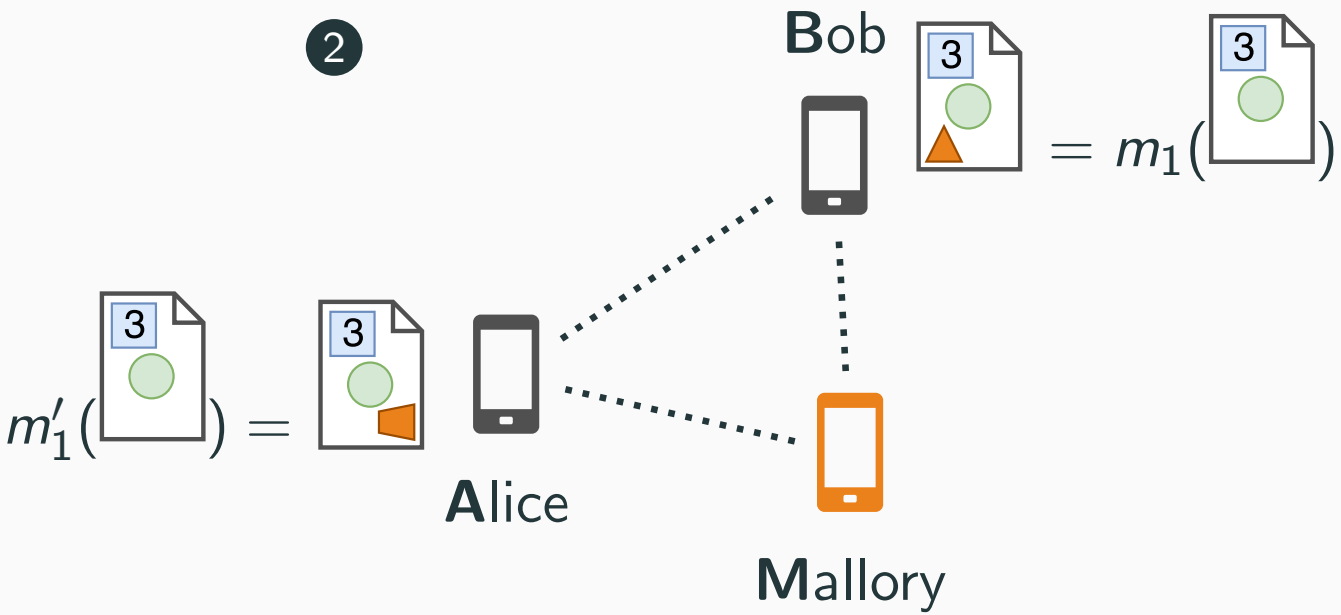
Threat to convergence



- Byzantine[1] adversary
  - **colluding malicious participants + network**
  - can **conceal divergences** using **equivocation**
- unlimited exchanges between honest participants

[1] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

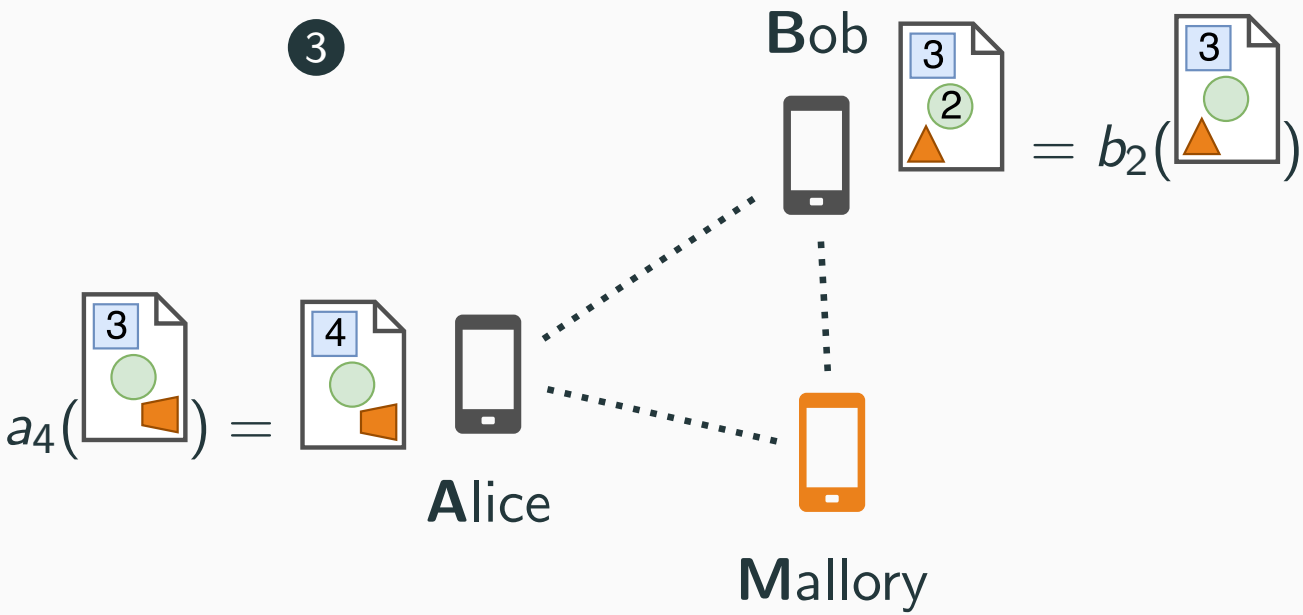
Threat to convergence



- Byzantine[1] adversary
  - **colluding malicious participants + network**
  - can **conceal divergences** using **equivocation**
- unlimited exchanges between honest participants

[1] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

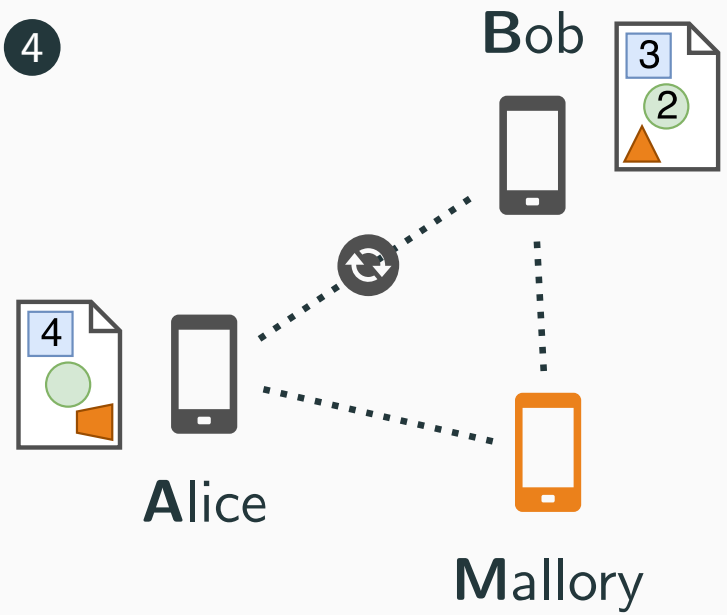
Threat to convergence



- Byzantine[1] adversary
  - **colluding malicious participants + network**
  - can **conceal divergences** using **equivocation**
- unlimited exchanges between honest participants

[1] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

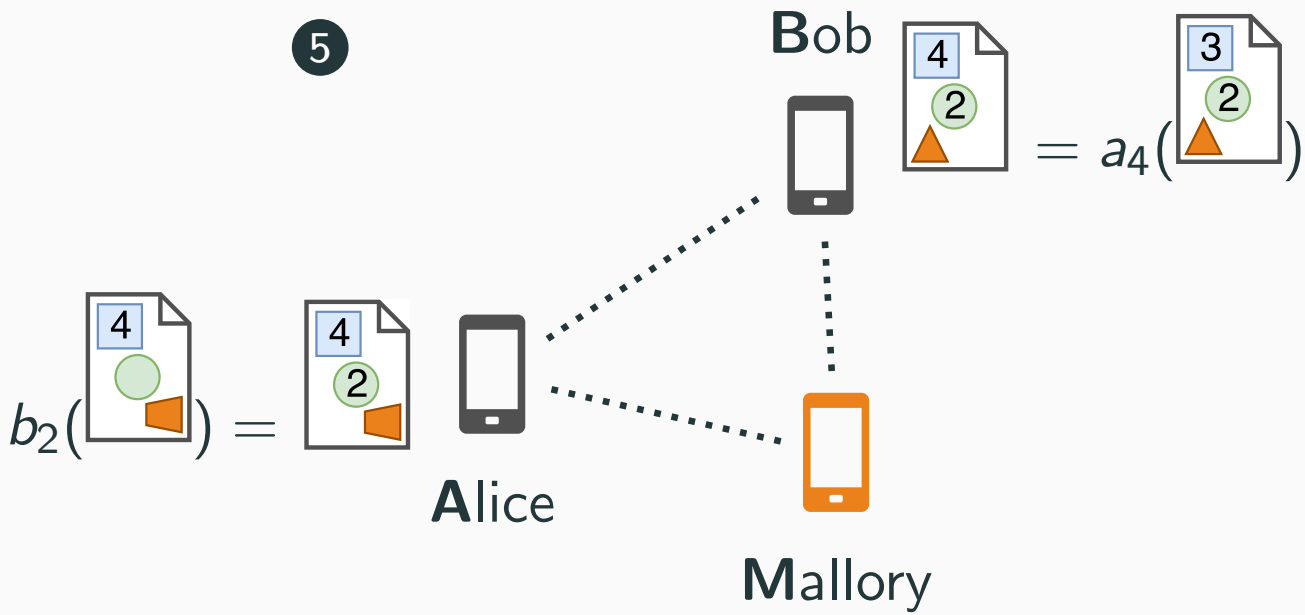
Threat to convergence



- Byzantine[1] adversary
  - **colluding malicious participants + network**
  - can **conceal divergences** using **equivocation**
- unlimited exchanges between honest participants

[1] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

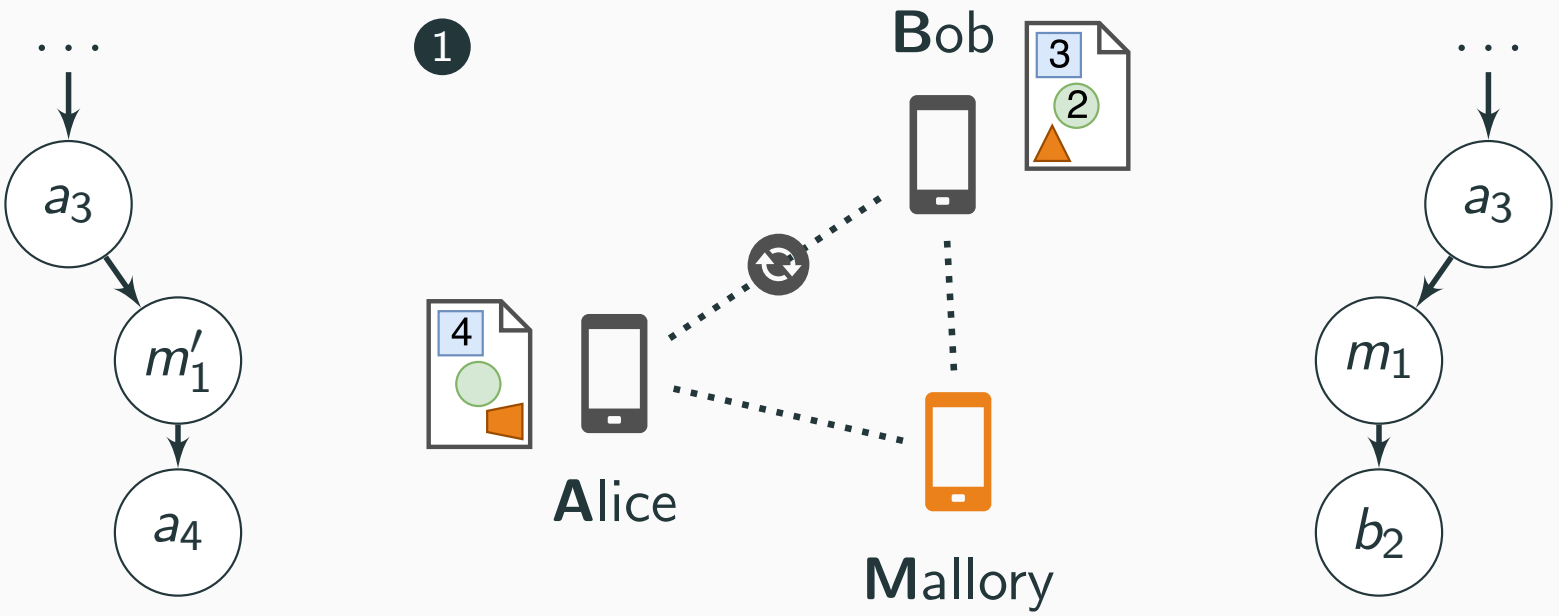
Threat to convergence



- Byzantine[1] adversary
  - **colluding malicious participants + network**
  - can **conceal divergences** using **equivocation**
- unlimited exchanges between honest participants

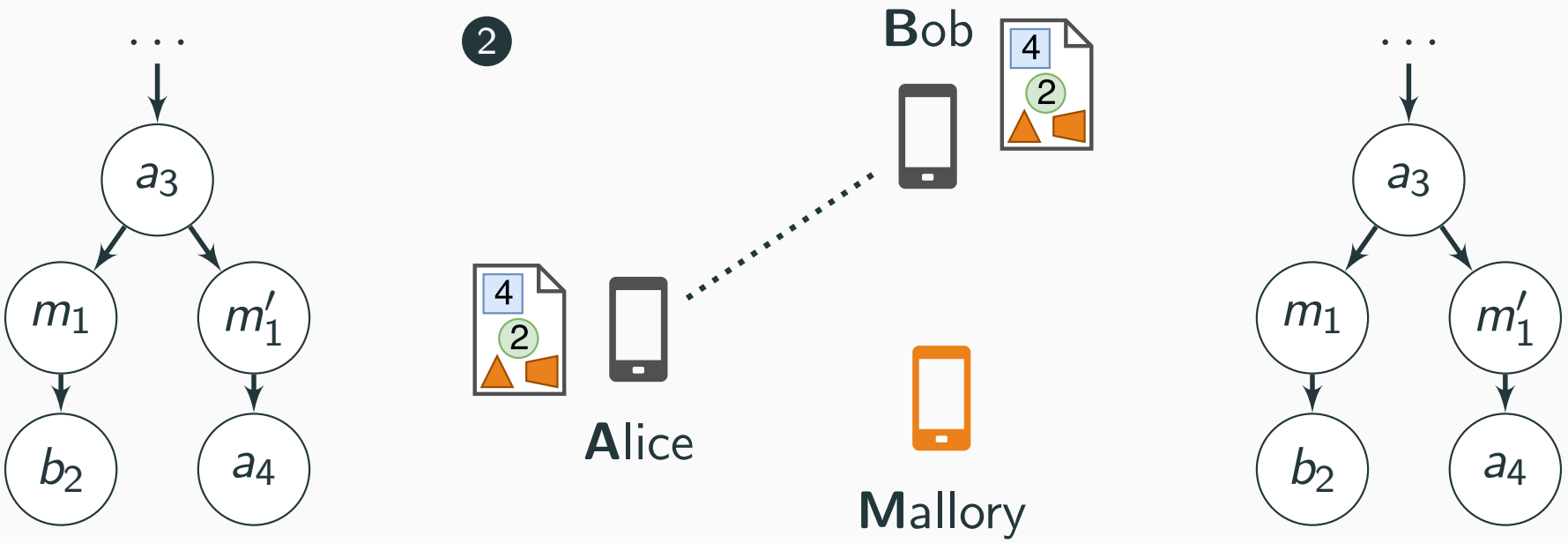
[1] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

Authenticated logs



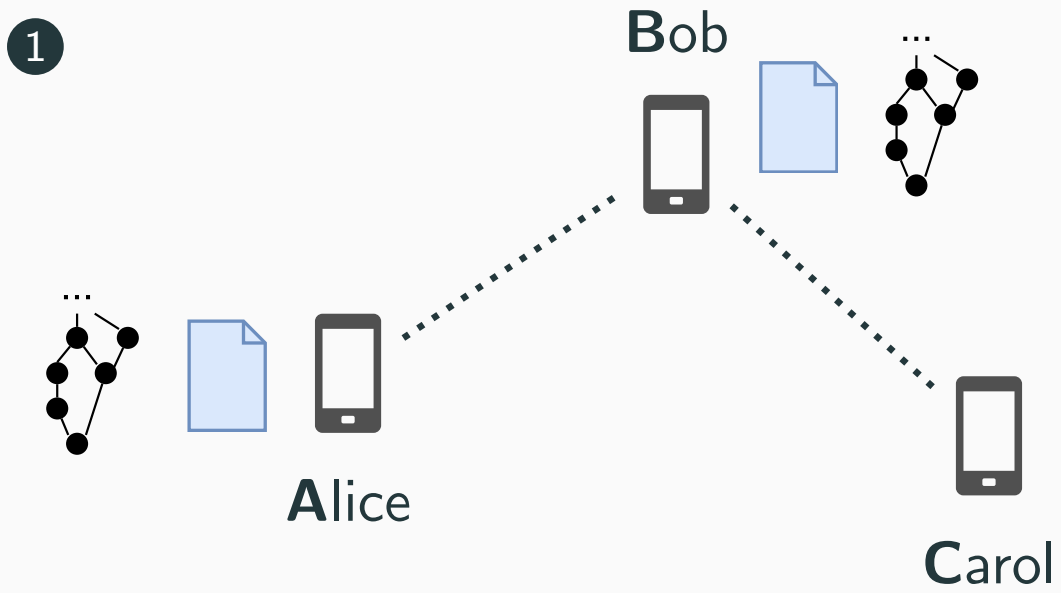
- **misbehaviors are accountable**
  - log entries are signed by their author
  - secured dependency tracking of operations thanks to hashes

Authenticated logs



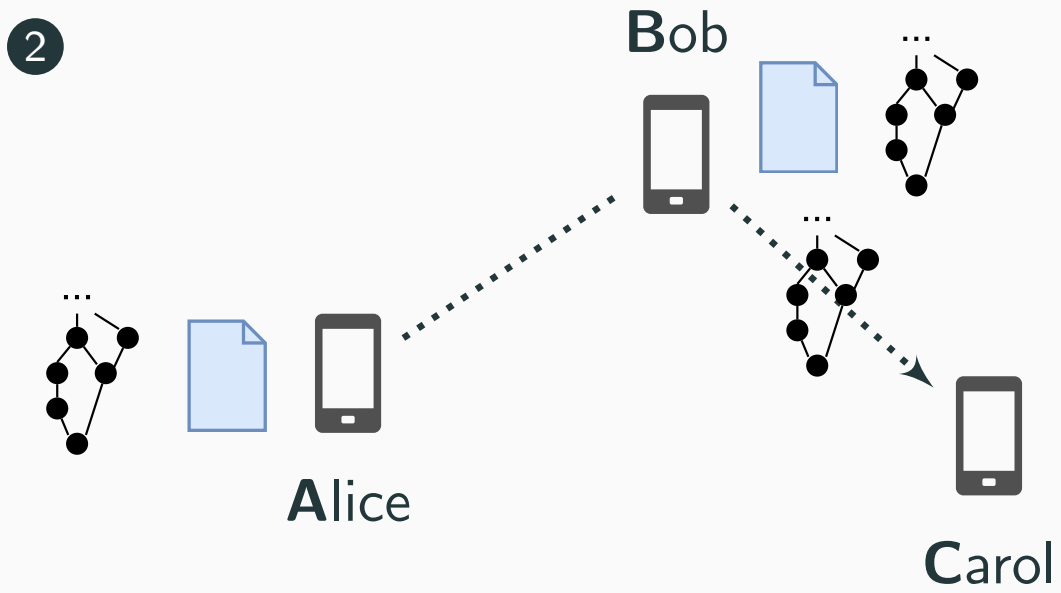
- **misbehaviors are accountable**
  - log entries are signed by their author
  - secured dependency tracking of operations thanks to hashes
- **strong convergence**
  - convergent logs $\Longrightarrow$ convergent documents

Overheads of authenticated logs



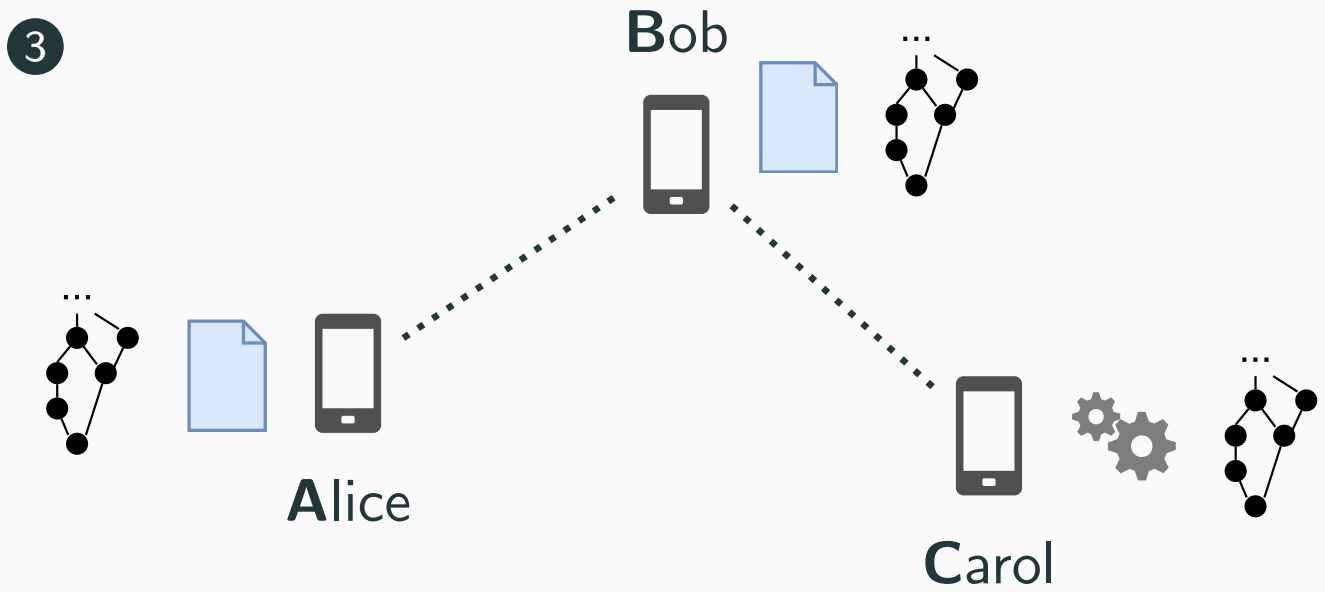
- participants **keep the entire log** for newcomers

Overheads of authenticated logs



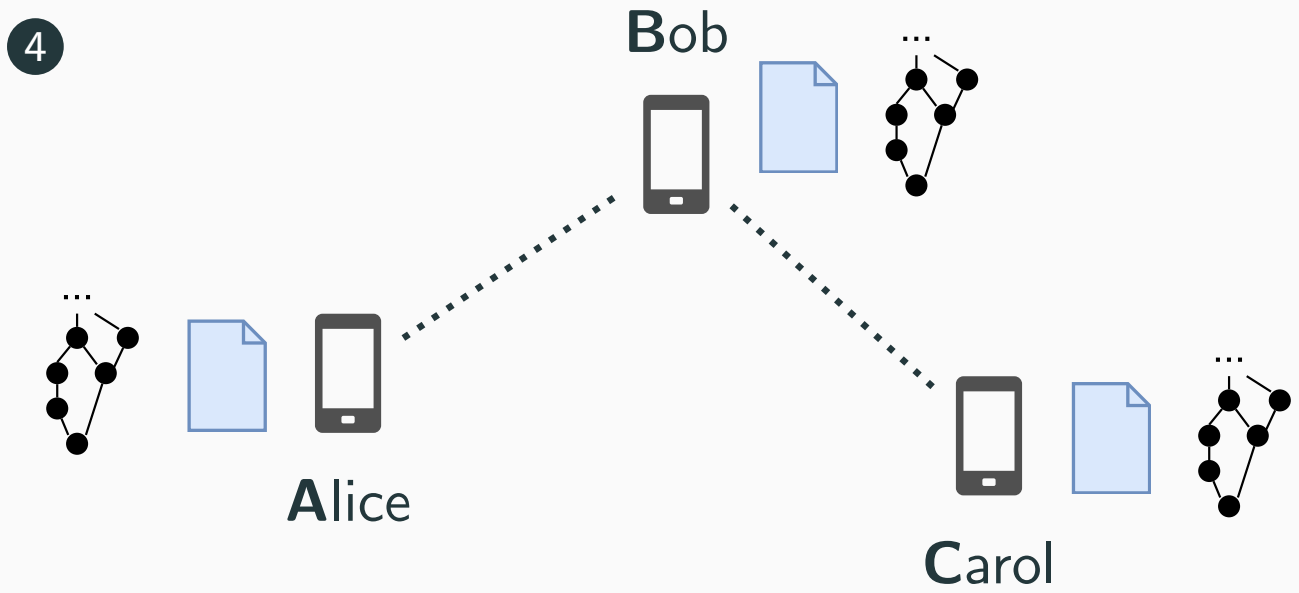
- participants **keep the entire log** for newcomers

Overheads of authenticated logs



- participants **keep the entire log** for newcomers
- **newcomers** verify and play the **entire log**

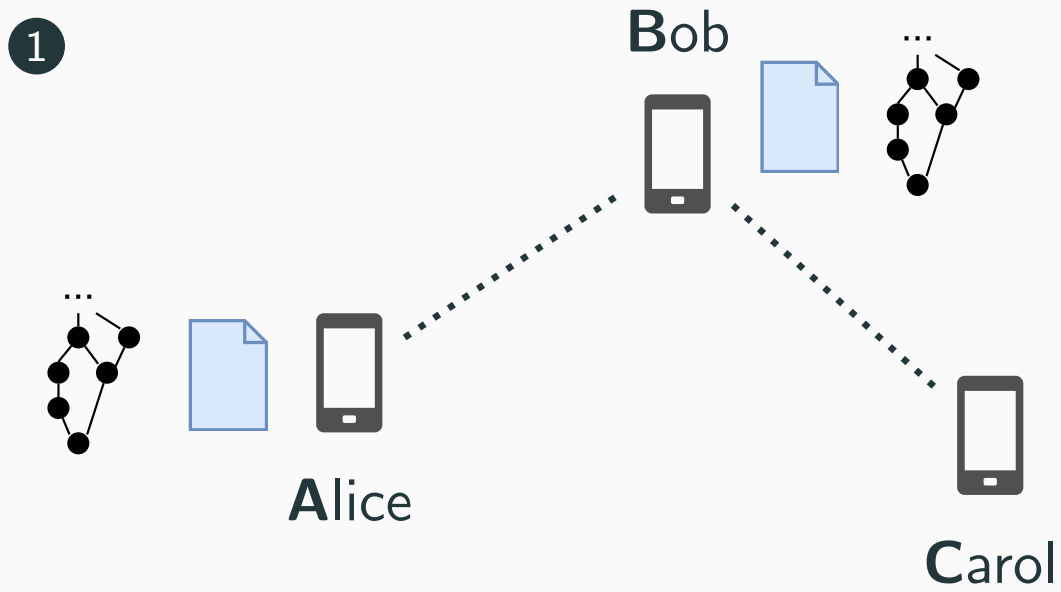
Overheads of authenticated logs



- participants **keep the entire log** for newcomers
- **newcomers** verify and play the **entire log**

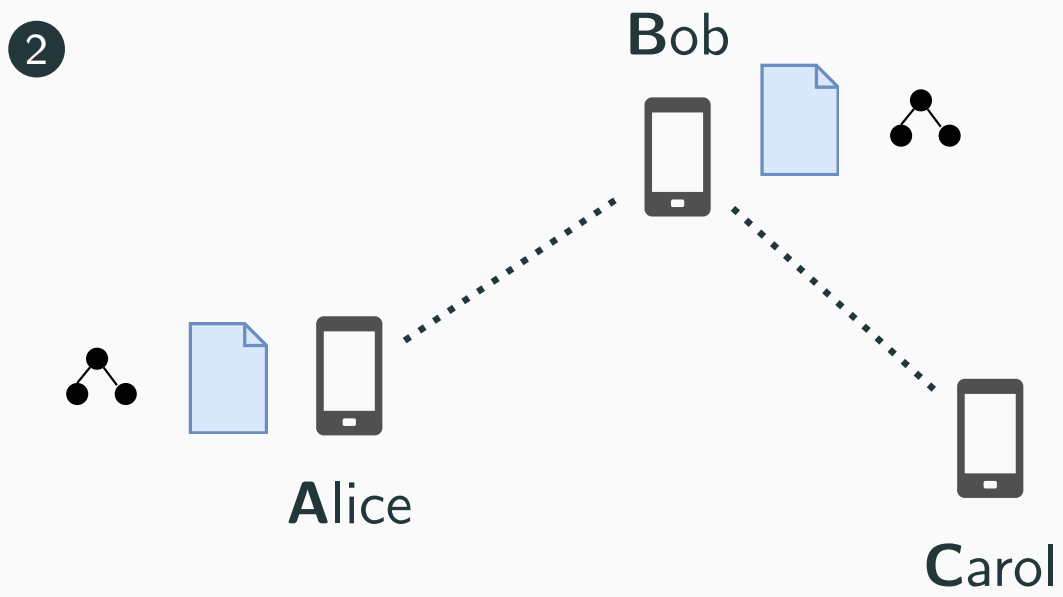
How to reduce overheads of authenticated logs in dynamic groups?

Overview of contributions



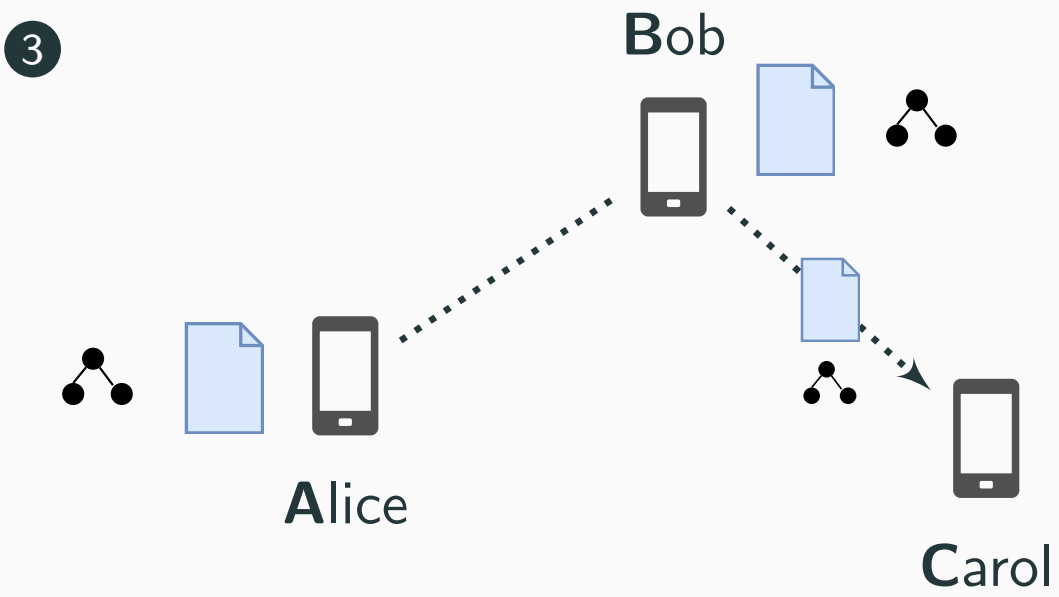
- participants **prune their log**

Overview of contributions



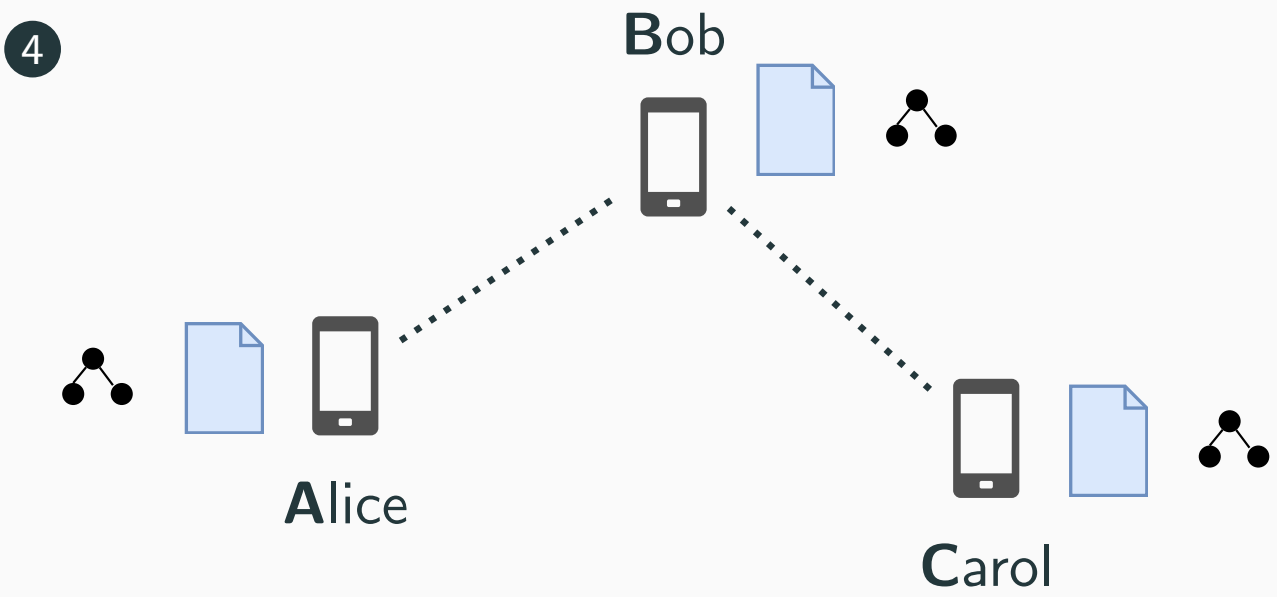
- participants **prune their log**

Overview of contributions



- participants **prune their log**
- newcomers **retrieve a document and its pruned log**
  1. verify the **consistency** of the pruned log
  2. authenticate the document using the pruned log

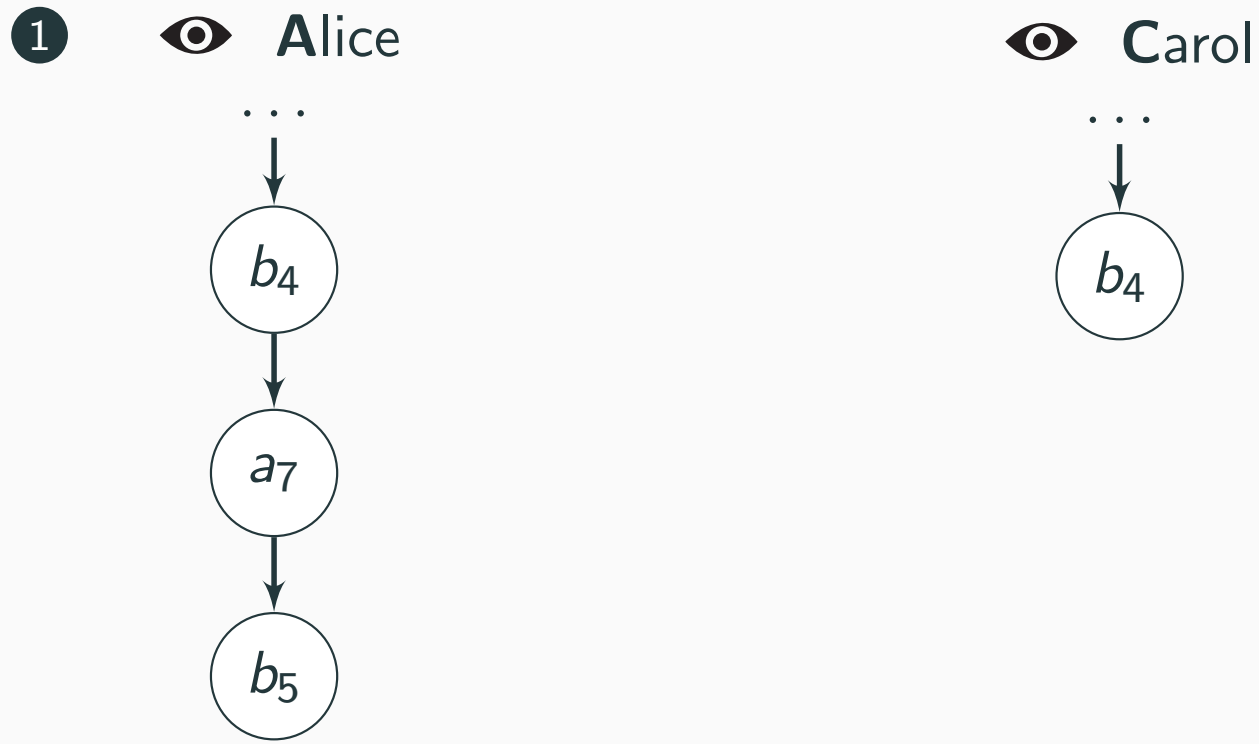
Overview of contributions



- participants **prune their log**
- newcomers **retrieve a document and its pruned log**
  1. verify the **consistency** of the pruned log
  2. authenticate the document using the pruned log

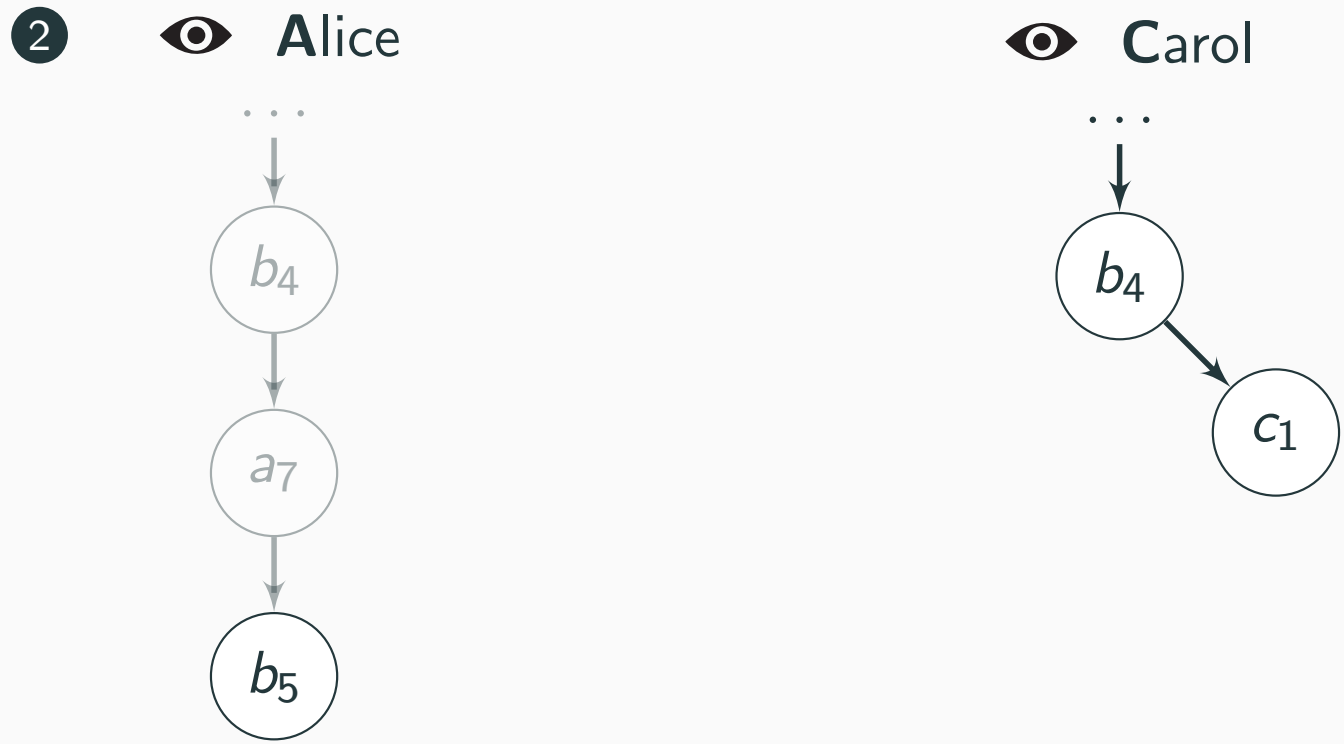
How to prune a log without threatening convergence?

Pruning issue



- operations **cannot be arbitrarily dropped**
  - operations are required to verify a given operation
- concurrency plays an important role

Pruning issue



- operations **cannot be arbitrarily dropped**
  - operations are required to verify a given operation
- concurrency plays an important role

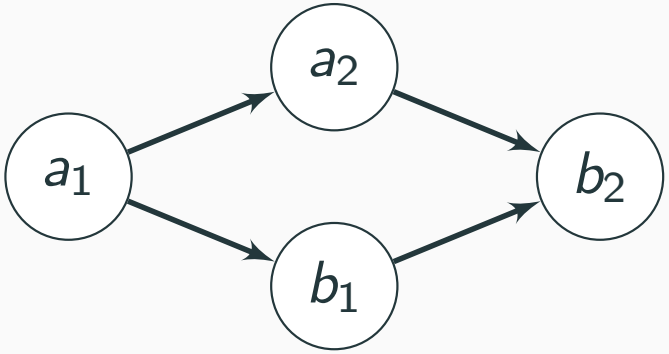
Stability

Definition

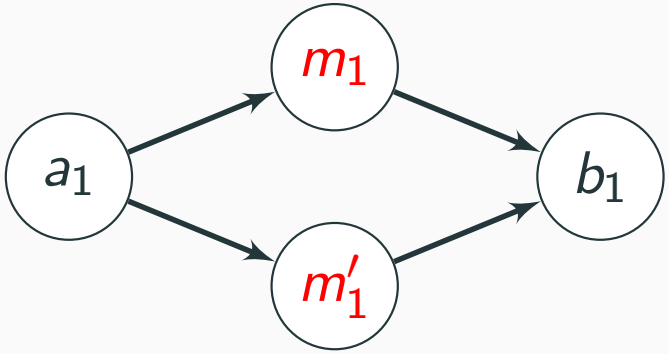
An operation is stable once it is no longer possible to accept a concurrent operation to it.

- primitive to define which operations can be dropped
    - stable part can be pruned
- necessity of **limiting concurrencies**

Consistency models and Causal consistency



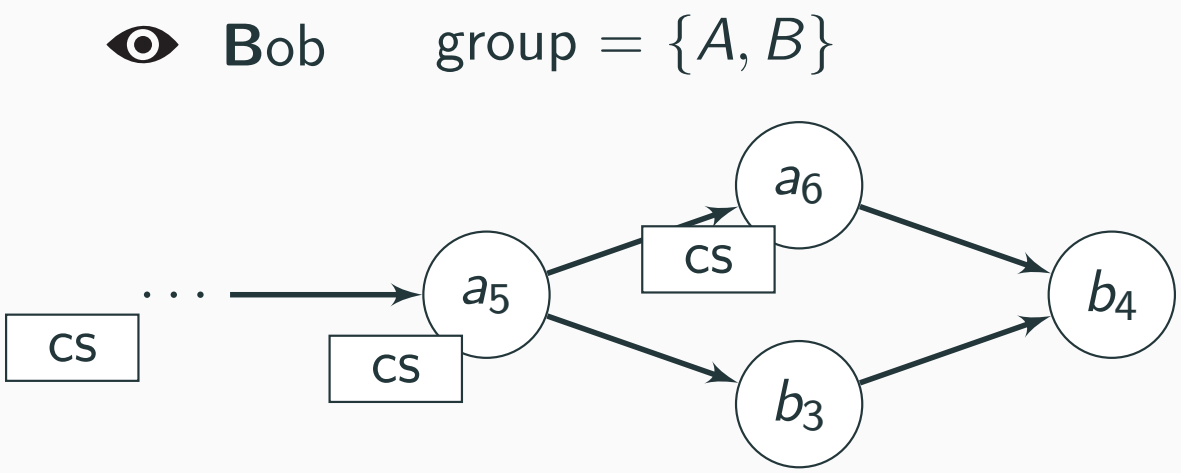
Causal consistent log



Non causal consistent log

- logs respect a **consistency model**
  - how operations can be related
- Causal consistency
  - Participants **linearly order** their operations

Causal Stability[2]



Definition

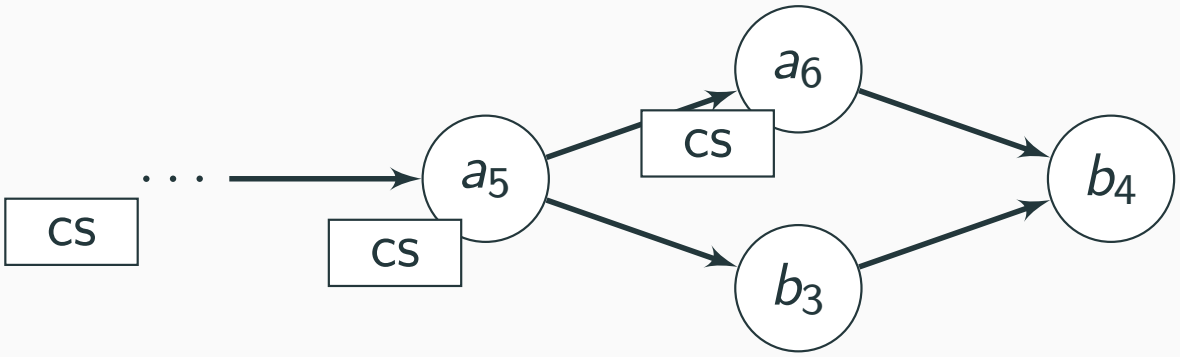
An operation is *causally stable* (cs) once all participants of the group have observed it.

---

[2] Carlos Baquero et al. Making Operation-Based CRDTs Operation-Based. In *Distributed Applications and Interoperable Systems - 14th IFIP WG 6.1 International Conference, DAIS 2014, Held as Part of the 9th International Federated Conference on Distributed Computing Techniques*. Springer, 2014 .

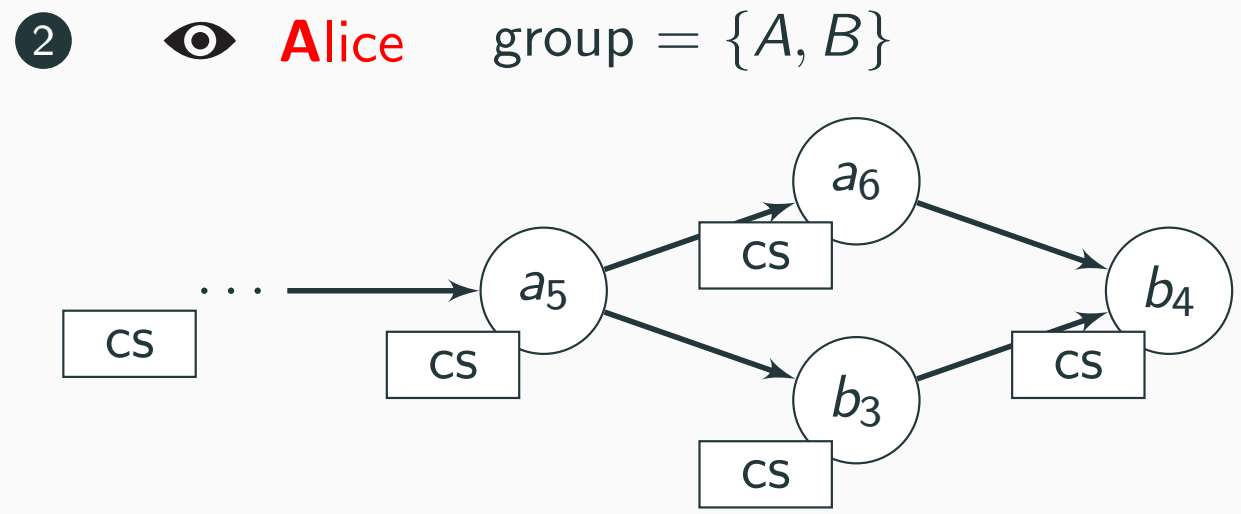
Limitations of Causal Stability

❶ 👁 **B**ob group $= \{A, B\}$



Issues

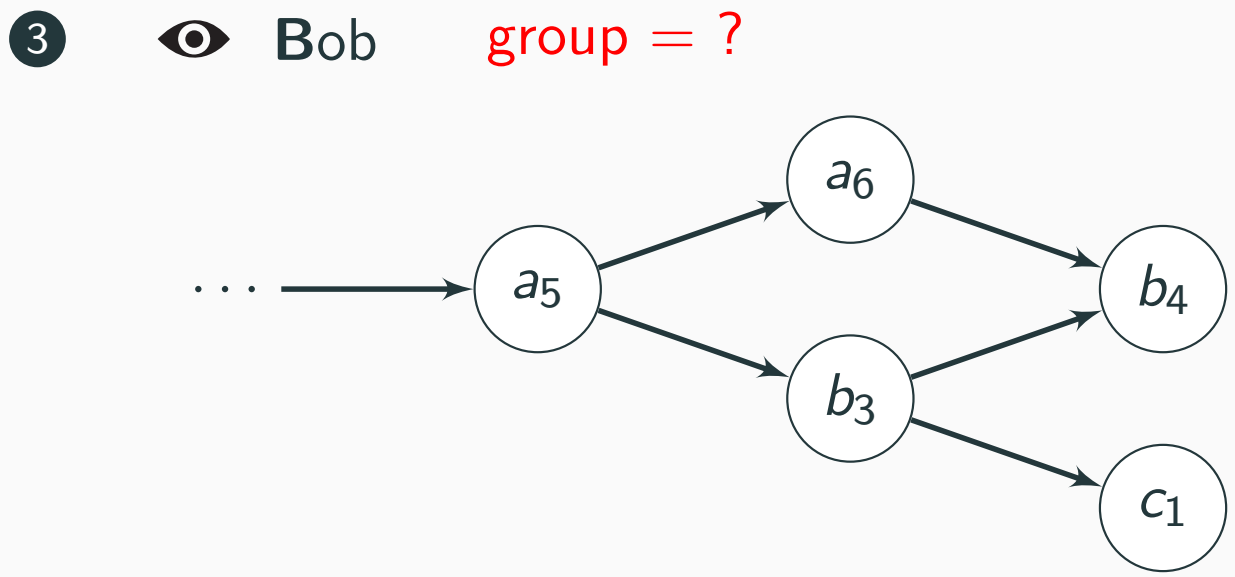
Limitations of Causal Stability



Issues

- convergent logs do not imply **convergent stability**

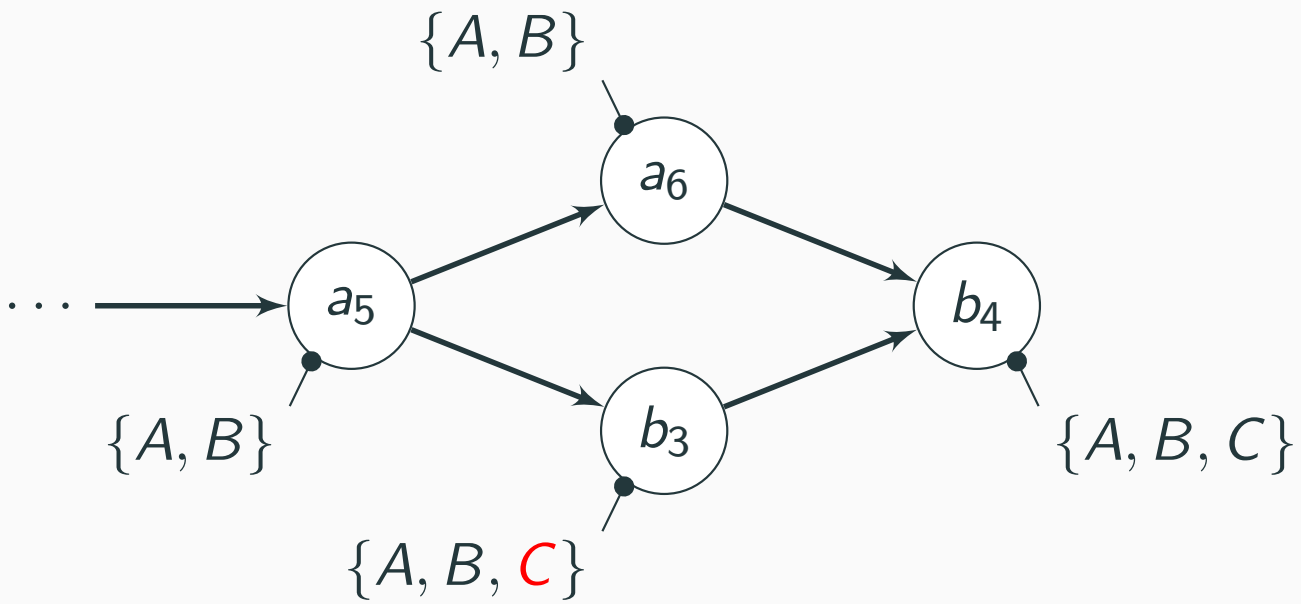
Limitations of Causal Stability



Issues

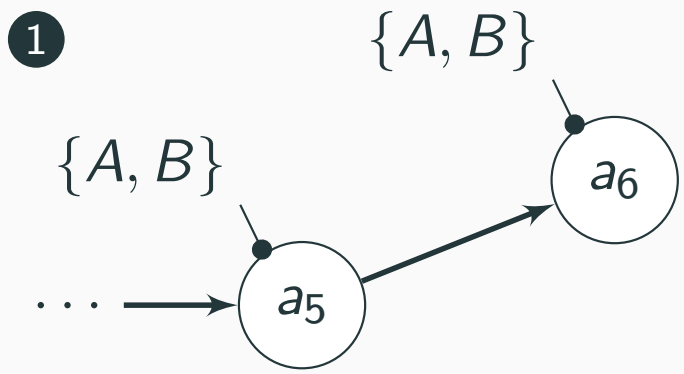
- convergent logs do not imply **convergent stability**
- does not fit **dynamic groups**

Membership tracking



- Operations embed a list of the group members
- Dependencies of an operation must include its author

Contribution: Provable causal stability



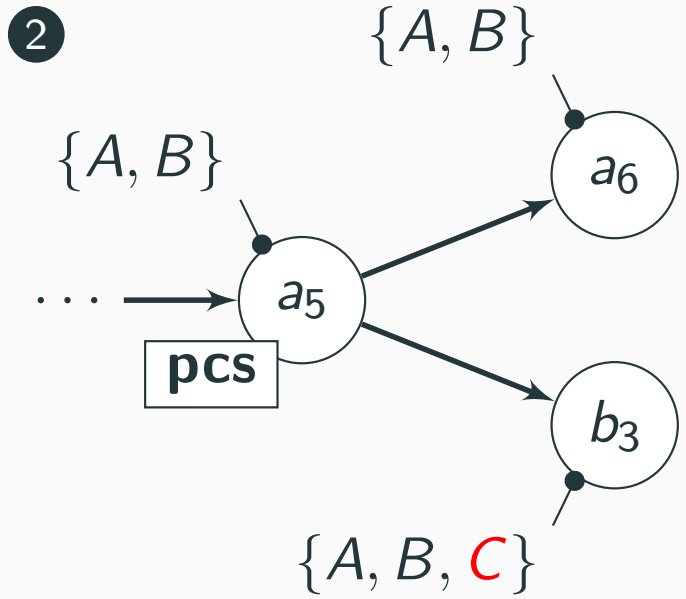
Definition

An operation is provably causally stable (pcs) once all required observers have provably observed it.

required observers =

+ participants at the moment of generation
+ participants concurrently invited

Contribution: Provable causal stability



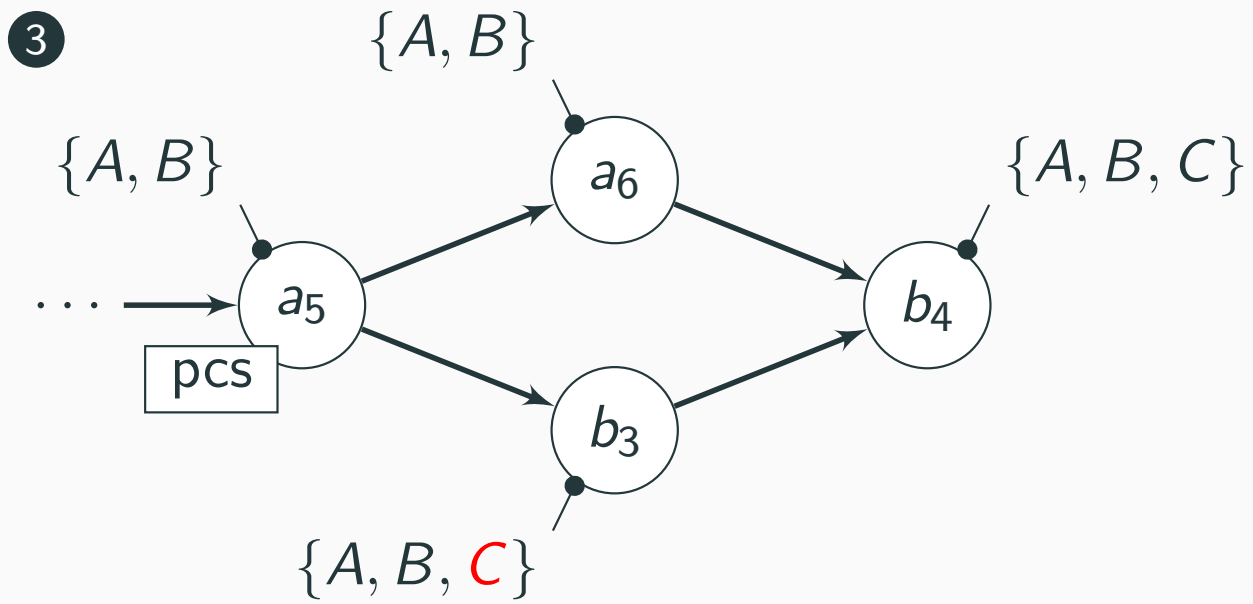
Definition

An operation is provably causally stable (pcs) once all required observers have provably observed it.

required observers =

+ participants at the moment of generation
+ participants concurrently invited

Contribution: Provable causal stability



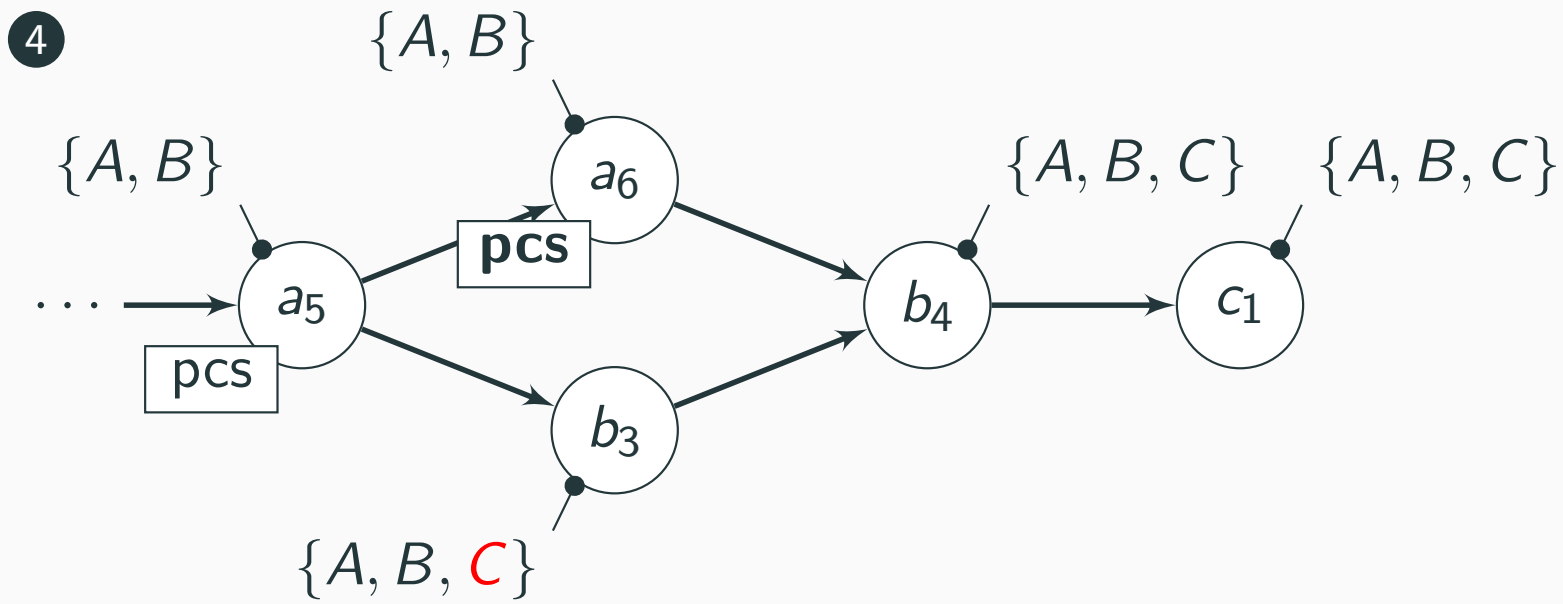
Definition

An operation is provably causally stable (pcs) once all required observers have provably observed it.

required observers =

+ participants at the moment of generation
+ participants concurrently invited

Contribution: Provable causal stability



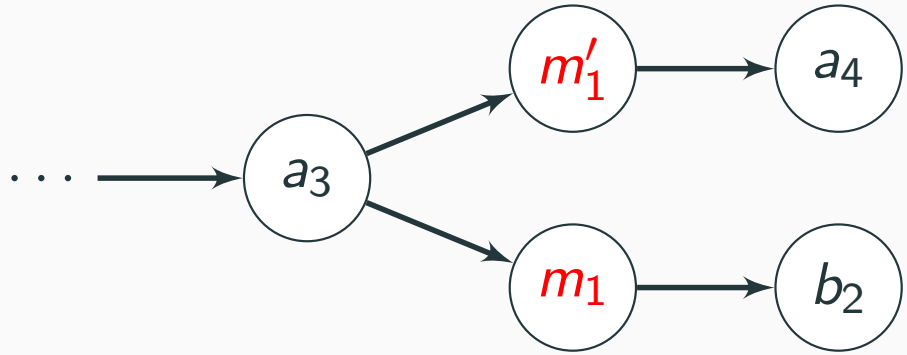
Definition

An operation is provably causally stable (pcs) once all required observers have provably observed it.

required observers =

+ participants at the moment of generation
+ participants concurrently invited

Malicious participants and causal consistency



- Malicious participants can issue **non-linear operations**
  - Causal consistency is not achievable

View-Fork-Join-Causal consistency[3]

VFJC-consistent VFJC-consistent

- honest participants linearly order their operations
- operations **cannot directly depend on non-linear** ones

[3] Prince Mahajan et al. Consistency, Availability, and Convergence. Technical report TR-11-22, The University of Texas at Austin, 2011.

View-Fork-Join-Causal consistency[3]

non VFJC-consistent

VFJC-consistent

- honest participants linearly order their operations
- operations **cannot directly depend on non-linear** ones

[3] Prince Mahajan et al. Consistency, Availability, and Convergence. Technical report TR-11-22, The University of Texas at Austin, 2011.

View-Fork-Join-Causal consistency[3]

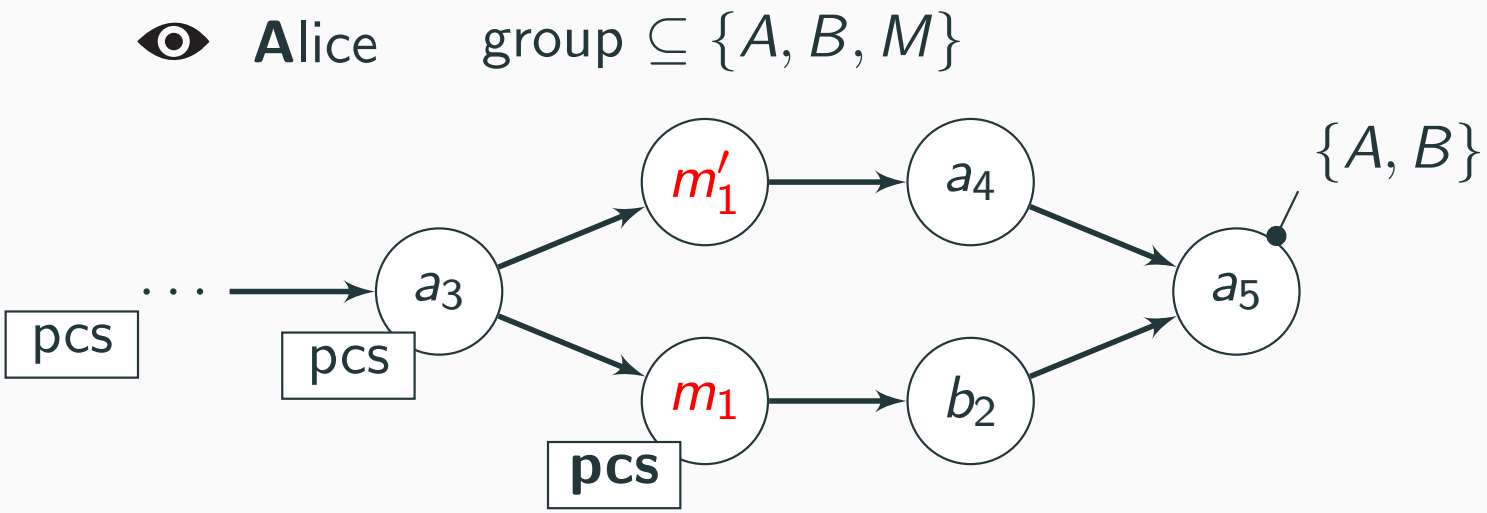
VFJC-consistent

VFJC-consistent

- honest participants linearly order their operations
- operations **cannot directly depend on non-linear** ones

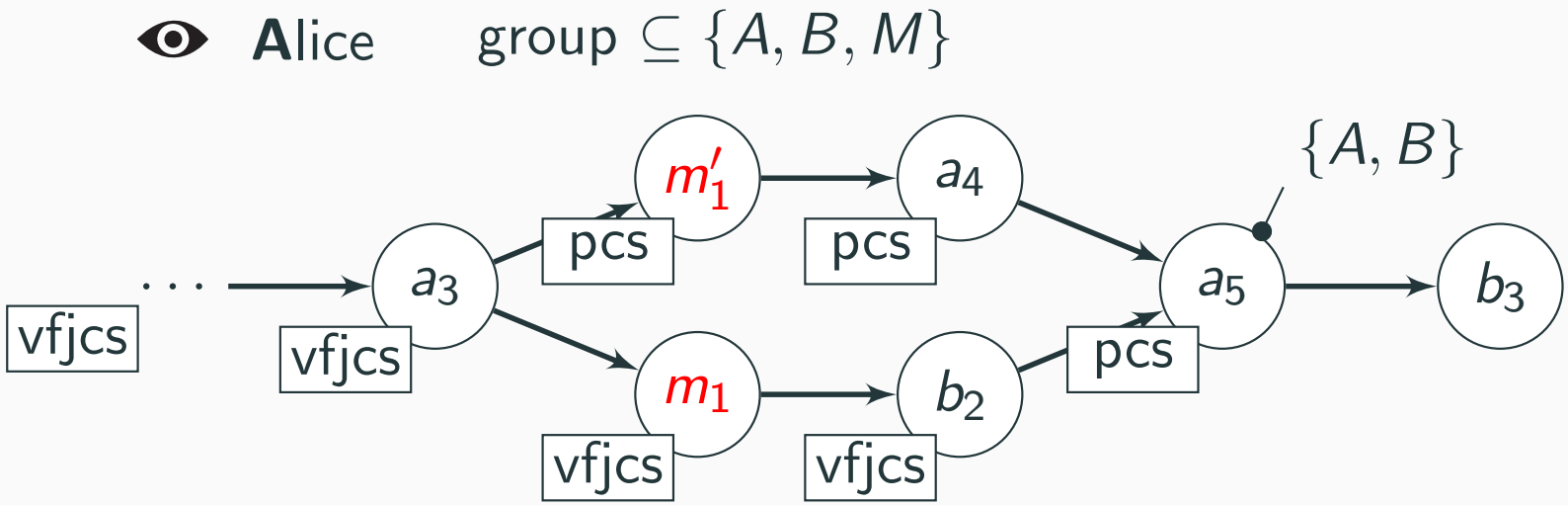
[3] Prince Mahajan et al. Consistency, Availability, and Convergence. Technical report TR-11-22, The University of Texas at Austin, 2011.

VFJC consistency



- **impossibility of** honestly accepting concurrent operations which are **non-linear with a provably causally stable** operation

Contribution: VFJC Stability



Definition

An operation is VFJC-stable (vfjcs) once all required observers have a provably causally stable operation which depends on.

- VFJC stability $\implies$ provable causal stability

Stability

Stability	CS	**PCS**	**VFJCS**	**VFJCS**
Consistency model	Causal	$\sim$ Causal	VFJC	**SVFJC**
Group membership	Static	Dynamic	Static	Dynamic
environment	Honest	Honest	Malicious	Malicious

CS: Causal Stability

PCS: Provable Causal Stability

VFJC: View-Fork-Join-Causal

VFJCS: View-Fork-Join-Causal Stability

SVFJC: Stabilizable View-Fork-Join-Causal

Conclusions

Contributions

- adaptation of **Causal Stability for dynamic groups**
- definition of **VFJC Stability**
- Stabilizable View-Fork-Join-Causal (VFJC) (not covered here)
- document **authentication based on a pruned log**

Perspectives

- exploring sub-group log pruning
- VFJC Stability as building block of **asynchronous Byzantine consensus**

References

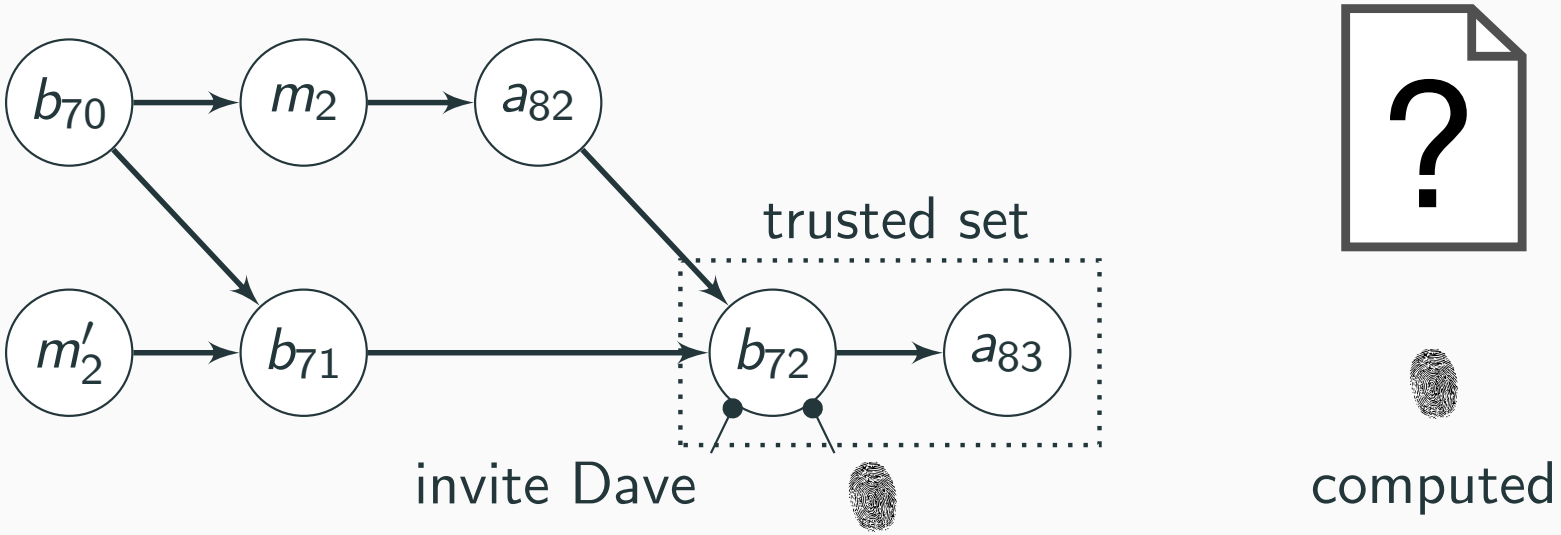
[1] Carlos Baquero et al. Making Operation-Based CRDTs Operation-Based. In *Distributed Applications and Interoperable Systems - 14th IFIP WG 6.1 International Conference, DAIS 2014, Held as Part of the 9th International Federated Conference on Distributed Computing Techniques*. Springer, 2014.

[2] Prince Mahajan et al. Consistency, Availability, and Convergence. Technical report TR-11-22, The University of Texas at Austin, 2011.

[3] Leslie Lamport et al. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, July 1982.

How to authenticate an untrusted snapshot?

Overview of document authentication



- participants embed a **document fingerprint in invitations**
  - participants verify fingerprints
- **computed** fingerprint = **embedded** fingerprint
  $\Longrightarrow$ **document considered authentic**
- a newcomer trusts a **fingerprint verified by her trusted set**