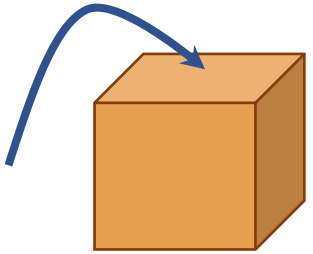
HOLISTIC VERIFICATION OF BLOCKCHAIN CONSENSUS

Nathalie Bertrand, Vincent Gramoli, Igor Konnov, Marijana Lazic,
Pierre Tholoniat, Josef Widder

Blockchain

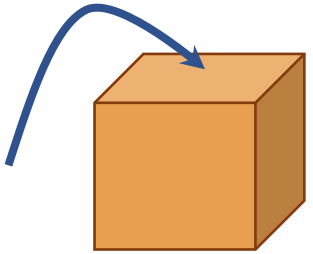
Tx1: Mallory gives all her coins to Alice



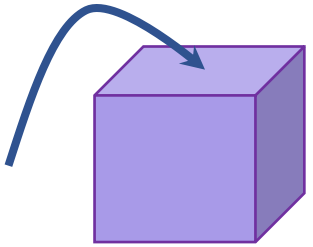
Double Spending

Mallory

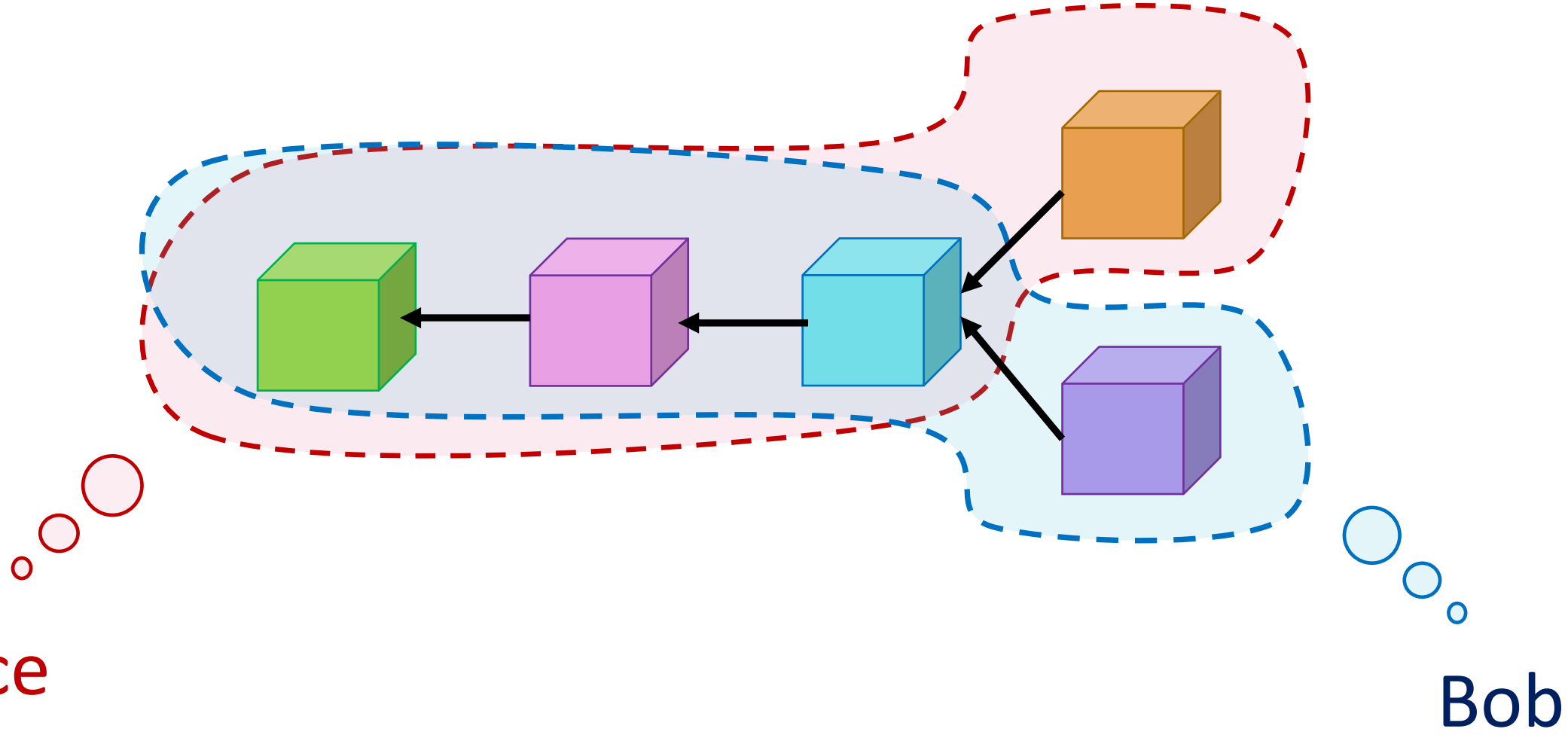
Tx1: Mallory gives all her coins to Alice



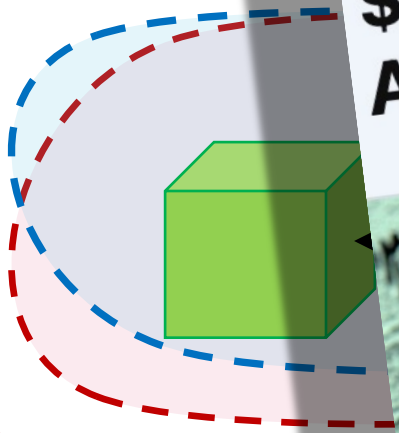
Tx2: Mallory gives all her coins to Bob



Loss of Assets



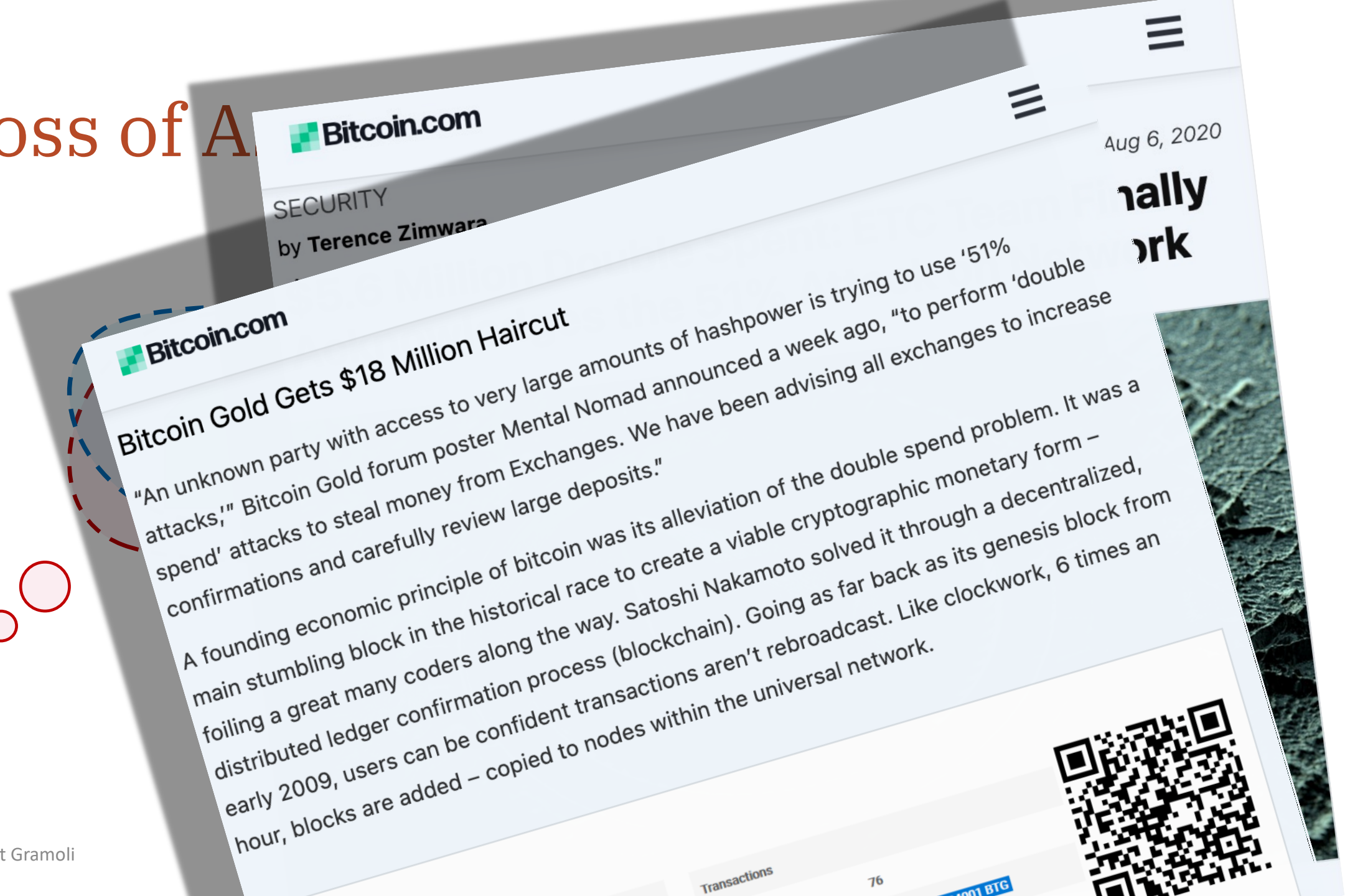
Loss of A



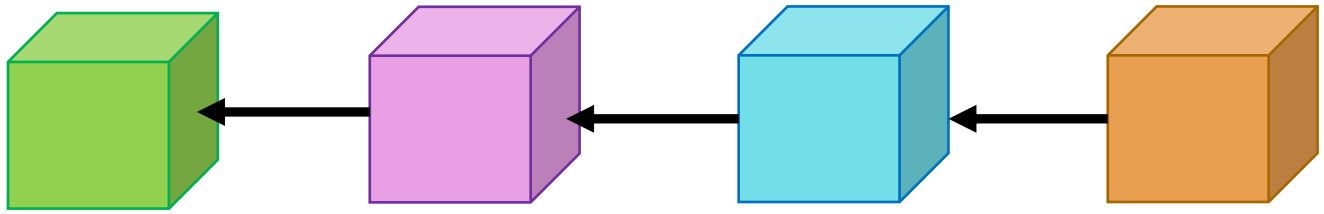
Alice

Loss of A

Alice



The Need for Byzantine Consensus



Alice

Bob

Impact of Human Errors

Red Belly Blockchain [IEEE S&P'21]

- Builds upon DBFT, a partially synchronous consensus protocol

- Can scale, its performance increasing with the system size

- Peaks at 660,000 transactions per second

- Deployed on 1000 machines across 4 continents

- Achieves 3 second finality (commit time)

Holistic Verification of DBFT

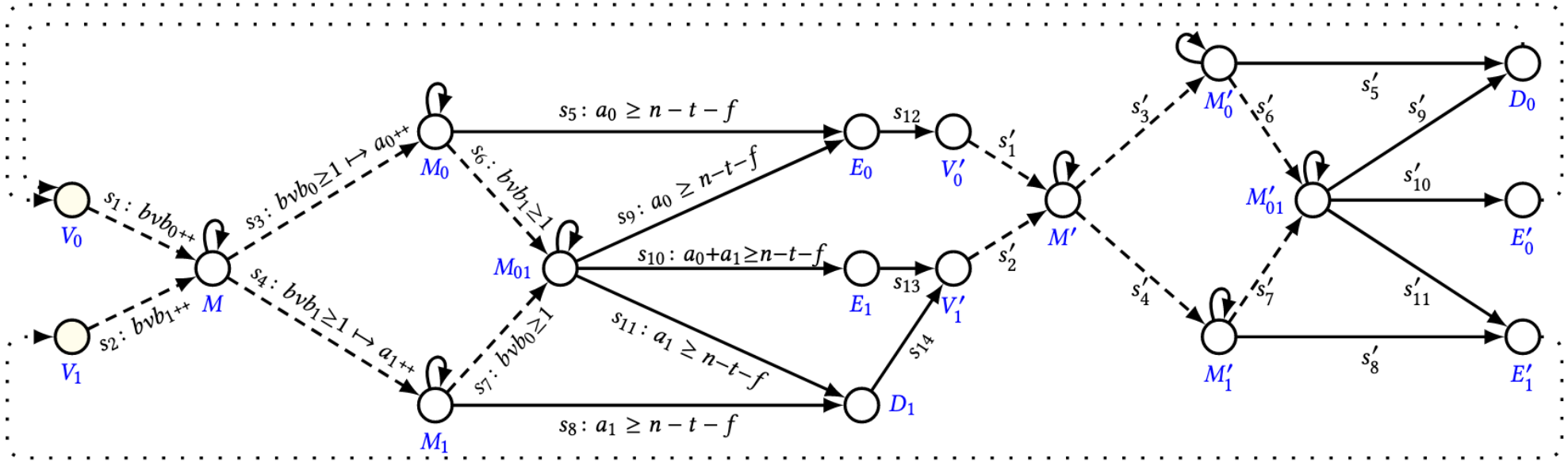
DBFT [NCA'18]

1. Fairness

```
1: Global scope variable:
2:    contestants ⊆ {0, 1}, set of binary values, initially ∅.

3: propose(est):
4:    r ← 0
5:    repeat:
6:       bv-broadcast(EST, ⟨est, i⟩)
7:       wait until (contestants ≠ ∅)
8:       broadcast(AUX, ⟨contestants, i⟩) → favorites
9:       wait until ∃c_1, ..., c_{n−t} : ∀1 ≤ j ≤ n − t favorites[c_j] ≠ ∅
10:            ∧ (qualifiers ← ∪_{1≤j≤n−t} favorites[c_j]) ⊆ contestants
11:      if qualifiers = {v} then
12:         est ← v
13:         if v = (r mod 2) then decide(v)
14:      else est ← (r mod 2)
15:      r ← r + 1
```

2. Alg. to TA [OPODIS'17]



3. DBFT TA Specification

Byzantine Consensus Problem

$$\forall R \in \mathbb{N}, \forall R' \in \mathbb{N} \left(\Diamond \kappa[D_v, R] \neq 0 \Rightarrow \Box \kappa[D_{1-v}, R'] = 0 \right) \quad (Agree_v)$$

$$\forall R \in \mathbb{N} \left(\kappa[V_v, 1] = 0 \Rightarrow \Box \kappa[D_v, R] = 0 \right) \quad (Valid_v)$$

$$\exists R \in \mathbb{N} \left(\Box \left(\kappa[E_0, R] = 0 \wedge \kappa[E_1, R] = 0 \right) \vee \Box \left(\kappa[E'_0, R] = 0 \wedge \kappa[E'_1, R] = 0 \right) \right) \quad (Term)$$

3. LTL Problem Specification

ByMC [POPL 2017]

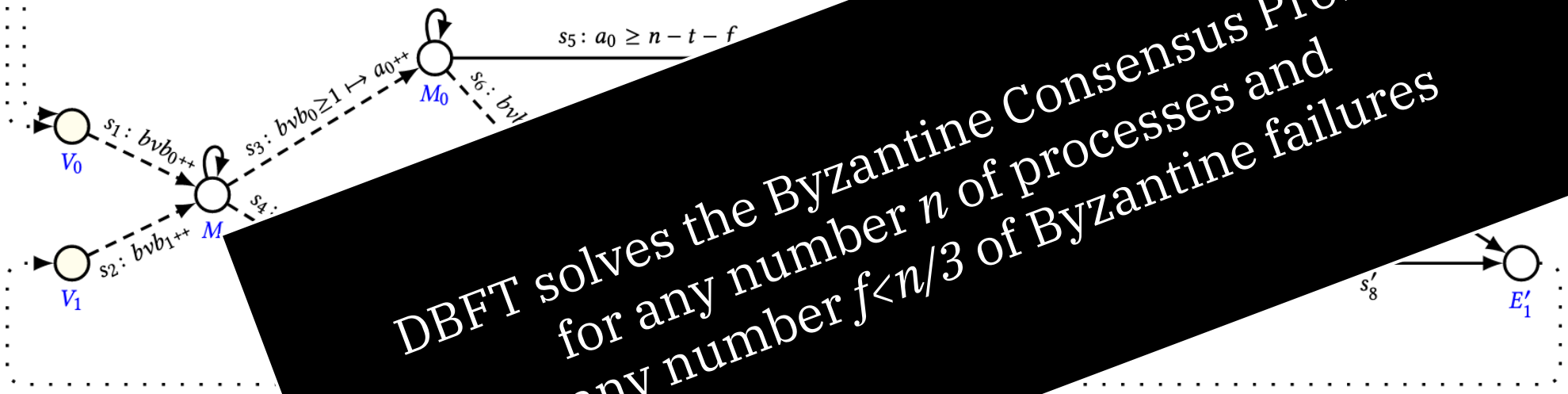
Holistic Verification of DBFT

DBFT [NCA'

```
1: Global scope variable:
2:    contestants ⊆ {0, 1}, set of binary values, initially ∅.

3: propose(est):
4:    r ← 0
5:    repeat:
6:       bv-broadcast(EST, ⟨est, i⟩)
7:       wait until (contestants ≠ ∅)
8:       broadcast(AUX, ⟨contestants, i⟩) → favorites
9:       wait until ∃c_1, ..., c_{n−t} : ∀1 ≤ j ≤ n − t favorites[c_j] ≠ ∅
10:            ∧ (qualifiers ← ∪_{1≤j≤n−t} favorites[c_j]) ⊆ contestants
11:      if qualifiers = {v} then
12:         est ← v
13:         if v = (r mod 2) then decide(v)
14:      else est ← (r mod 2)
15:      r ← r + 1
```

2. Alg. to TA [OPODIS'17]



DBFT solves the Byzantine Consensus Problem for any number n of processes and for any number $f<n/3$ of Byzantine failures

3. DBFT TA Specification

Byzantine (

$$\forall R \in \mathbb{N}, \forall R' \in \ldots \left(\ldots [\mathcal{D}_{1-v}, R'] = 0\right) \quad (Agree_v)$$

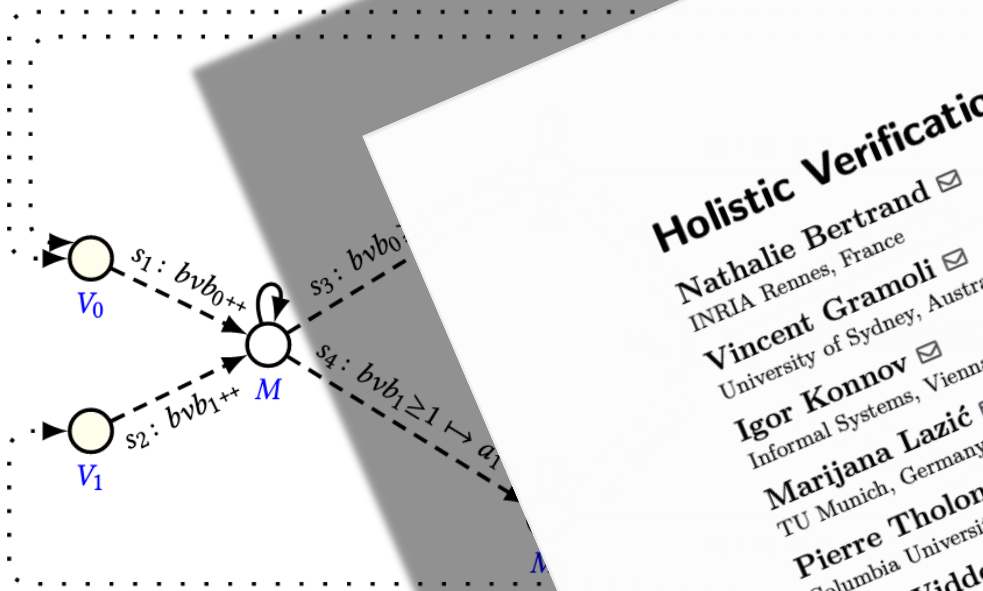
$$\forall R \in \mathbb{N} \left(\kappa[V_v, 1] \ldots [\mathcal{D}_v, R] = 0\right) \quad (Valid_v)$$

$$\exists R \in \mathbb{N} \left(\Box\left(\kappa[E_0, R] \ldots 0 \wedge \kappa[E_1, R] = 0\right) \vee \Box\left(\kappa[E'_0, R] = 0 \wedge \kappa[E'_1, R] = 0\right)\right) \quad (Term)$$

3. LTL Problem Specification

ByMC [POPL 2017]

Holistic Verifica



Holistic Verification of Blockchain Consensus

Nathalie Bertrand ✉
INRIA Rennes, France
Vincent Gramoli ✉
University of Sydney, Australia
Igor Konnov ✉
Informal Systems, Vienna, Austria
Marijana Lazić ✉
TU Munich, Germany
Pierre Tholoniat ✉
Columbia University, New York, USA
Josef Widder ✉
Informal Systems, Vienna, Austria

Abstract

Blockchain has recently attracted the attention of the industry due, in part, to its ability to automate asset transfers. It requires distributed participants to reach a consensus on a block despite the presence of malicious (a.k.a. Byzantine) participants. Malicious participants exploit regularly weaknesses of these blockchain consensus algorithms, with sometimes devastating consequences. In fact, these weaknesses are quite common and are well illustrated by the flaw in various blockchain consensus algorithms. Paradoxically, until now, no blockchain consensus has been holistically verified using model checking.

In this paper, we remedy this paradox by model checking for the first time a blockchain consensus used in industry. We propose a holistic approach to verify the consensus algorithm of the Red Belly Blockchain [20], for any number n of processes and any number $f < n/3$ of Byzantine processes. We decompose directly the algorithm pseudocode in two parts—an inner broadcast algorithm and an outer decision algorithm—each modelled as a threshold automaton [36], and we formalize their expected properties in linear-time temporal logic. We then automatically check the inner ...sting algorithm, under a carefully identified fairness assumption. For the verification of the ...m, we simplify the model of the inner algorithm by relying on its checked properties. ...ly verify not only the safety properties of the Red Belly Blockchain consensus ...than 70 seconds.

... Distributed algorithms → ... outing methodologies → Distributed algorithms
... logic

...irness

```
1: Global scope variable:
2:    contestants ⊆ {0, 1}, set of binary values, initially ∅.

3: propose(est):
4:    r ← 0
5:    repeat:
6:       bv-broadcast(EST, ⟨est, i⟩)
7:       wait until (contestants ≠ ∅)
8:       broadcast(AUX, ⟨contestants, i⟩) → favorites
9:       wait until ∃c_1, ..., c_{n−t} : ∀1 ≤ j ≤ n − t favorites[c_j] ≠ ∅
10:            ∧ (qualifiers ← ∪_{1≤j≤n−t} favorites[c_j]) ⊆ contestants
11:      if qualifiers = {v} then
12:         est ← v
13:         if v = (r mod 2) then decide(v)
14:      else est ← (r mod 2)
15:      r ← r + 1
```

2. Pseudocode to TA

...ecification

ByMC,
...OPL 2017]

Vincent.Gramoli@sydney.edu.au

THE UNIVERSITY OF SYDNEY

Relater Work

- Theorem prover were used to prove parts of Stellar (not its quorum system) [DISC'19]

- The safety of Byzantine Paxos was proved by refinements (not its liveness) [DISC'11]

- Symbolic model checkers proved algorithms for fixed number of processes (n=10) [Dist. Comp. 2011]

- TLA+ model checker TLC was used for consensus with benign faults [RP'19]

- Bosco is a fast-path for consensus whose safety was proven with ByMC, but it needs to rely on another Byzantine consensus protocol [OPODIS'17]