

Διάλεξη 14: Ατομική ΚΚΜ Εγγραφής/Ανάγνωσης στην Παρουσία Σφαλμάτων

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι



Τι θα δούμε σήμερα

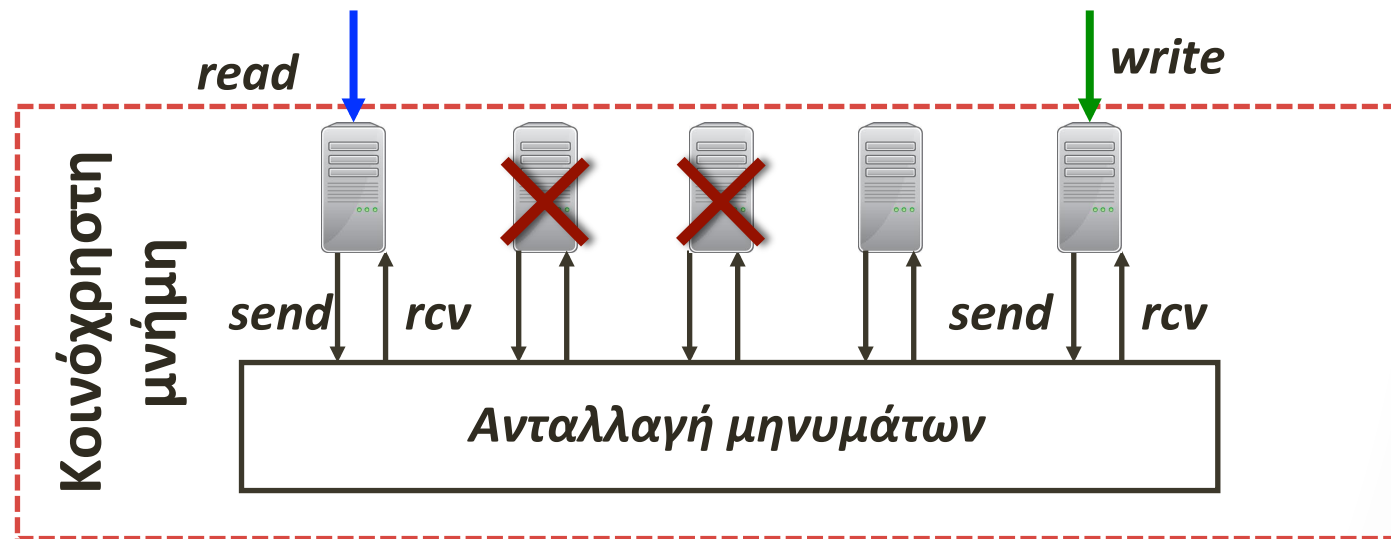
- Σφάλματα Κατάρρευσης
- Αλγόριθμος SWMR (ΜΕΠΑ) Ατομικής ΚΚΜ στην παρουσία σφαλμάτων κατάρρευσης (ABD - Attiya, Bar-Noy, Dolev)
- Αλγόριθμος MWMR (ΠΕΠΑ) Ατομικής ΚΚΜ στην παρουσία σφαλμάτων κατάρρευσης

Σφάλματα Κατάρρευσης

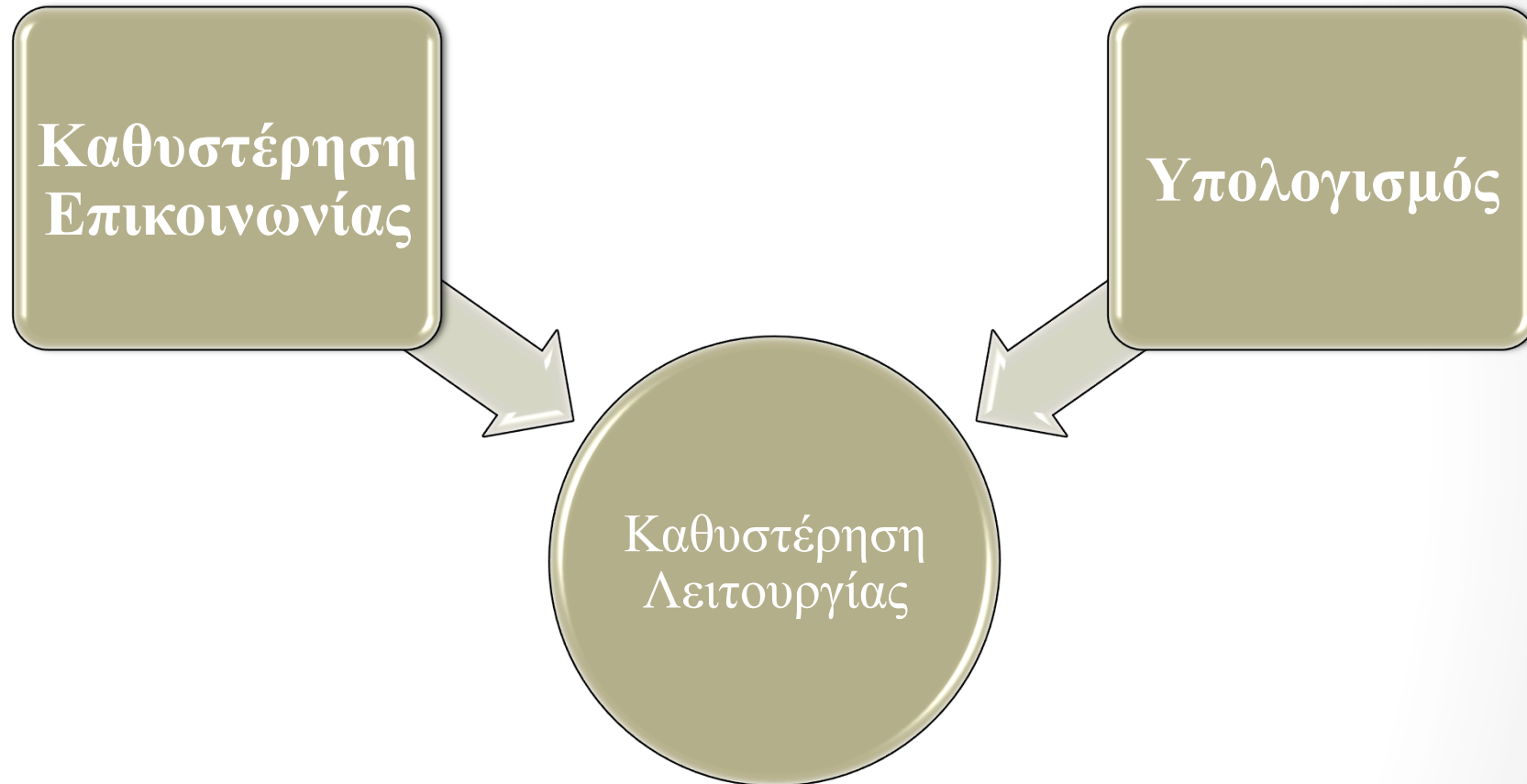
- Υποθέτουμε ότι κάθε επεξεργαστής p έχει ένα γεγονός $fail_p$
- Λέμε ότι ένας επεξεργαστής p **καταρρέει σε μια εκτέλεση α** εαν
 - Το γεγονός $fail_p$ περιέχεται στο α
 - Κανένα υπολογιστικό βήμα από τον p δεν εμφανίζεται στον α μετά από το $fail_p$
- Ένας επεξεργαστής μπορεί να καταρρεύσει κατά τη διάρκεια αποστολής μηνυμάτων
 - Έτσι ένα υποσύνολο των μηνυμάτων μπορεί να σταλεί
- Το σύνολο των επεξεργαστών που καταρρέουν δεν είναι γνωστό στους επεξεργαστές εκ των προτέρων

Ανεκτικότητα Σφαλμάτων στην ΚΚΜ

- **Πλεονασμός:** κράτα αντίγραφα της μνήμης σε πολλούς επεξεργαστές
 - Διασφαλίζει τη προσβασιμότητα και ζωτικότητα της ΚΚΜ παρά την ύπαρξη σφαλμάτων κατάρρευσης



Πολυπλοκότητα Αλγορίθμων



Υλοποίηση ΚΚΜ

- Για να υλοποιήσουμε ΚΜΜ πρέπει να προσδιορίσουμε τον αλγόριθμο 3 λειτουργιών
 - Εγγραφή
 - Ανάγνωση
 - Εξυπηρέτηση από διαχειριστές αντιγράφων
- Επομένως θεωρούμε 3 σύνολα διεργασιών
 - Εγγραφείς: **αλλάζουν την τιμή** μιας κοινόχρηστης μεταβλητής
 - Αναγνώστες: **διαβάζουν την τιμή** μιας κοινόχρηστης μεταβλητής
 - Εξυπηρετητές: **κρατούν αντίγραφο** μιας κοινόχρηστης μεταβλητής
- Ένας επεξεργαστής μπορεί να τρέχει και τις τρεις διεργασίες

Χαρακτηριστικά Υλοποίησης

- Πρέπει να προσδιορίζουμε
 - Μοντέλο Συγχρονισμού (σύγχρονο ή ασύγχρονο)
 - Μοντέλο Ταυτοχρονισμού
 - Μοντέλο Σφαλμάτων
 - Συνθήκη Συνέπειας
- Μοντέλο Ταυτοχρονισμού
 - SWSR
 - SWMR
 - MWMR
- Μοντέλου Σφαλμάτων
 - Είδος σφαλμάτων που υποστηρίζουμε
 - Αριθμός σφαλμάτων

Υλοποίηση Ατομικής ΚΚΜ

- Μοντέλο Συγχρονισμού: Ασύγχρονο
- Μοντέλο Ταυτοχρονισμού: SWMR (ΜΕΠΑ)
- Συνθήκη Συνέπειας: Ατομικότητα
- Μοντέλο Σφαλμάτων:
 - Κατάρρευση μειοψηφίας αντιγράφων ($(n/2)-1$)
 - Κατάρρευση οποιουδήποτε αριθμού αναγνωστών
 - Κατάρρευση εγγραφέα

Αλγόριθμος A

Write Protocol

- Στείλε μήνυμα `write(v)` σε ένα αντίγραφο και περίμενε απάντηση
- Μόλις λάβεις απάντηση επέστρεψε επιβεβαίωση και τερμάτισε

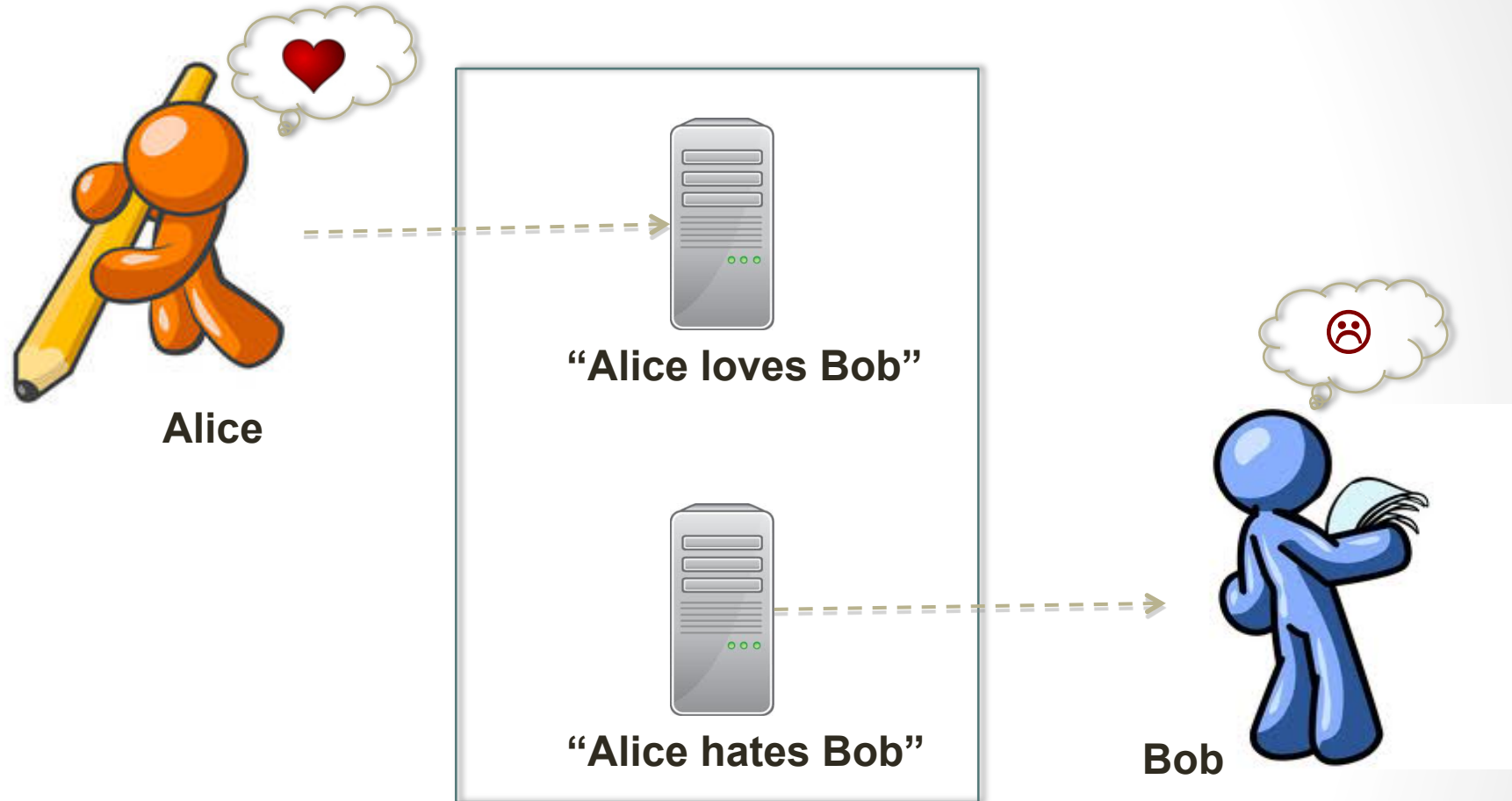
Read Protocol

- Στείλε μήνυμα `read()` σε ένα αντίγραφο και περίμενε απάντηση
- Μόλις λάβεις απάντηση επέστρεψε την τιμή που βρίσκεται μέσα στην απάντηση

Server Protocol

- Αν λάβεις μήνυμα `write(v)` άλλαξε την τιμή του αντιγράφου σου σε `v`
- Απάντησε σε όποιο μήνυμα παραλάβεις με `reply(v)` όπου `v` η τιμή του αντιγράφου σου.

Παράδειγμα Εκτέλεσης Αλγ. Α



Ανάλυση Αλγορίθμου A

- **Δεν Ικανοποιεί Νόμιμες Ακολουθίες:** ανάγνωση μπορεί να μην επιστρέψει την τιμή της τελευταίας εγγραφής
 - Επομένως δεν ικανοποιεί καμιά Συνθήκη Συνέπειας
- **Δεν είναι Ανεκτικός σε σφάλματα**
 - Σφάλμα ενός αντιγράφου μπορεί να εμποδίσει τον τερματισμό μιας λειτουργίας
- **Άρα ο Αλγόριθμος A είναι λανθασμένος**

Αλγόριθμος Β

Write Protocol

- Στείλε μήνυμα `write(v)` σε **όλα τα αντίγραφα** και περίμενε **απάντηση από την πλειοψηφία**
- Μόλις λάβεις απάντησεις επέστρεψε επιβεβαίωση και τερμάτισε

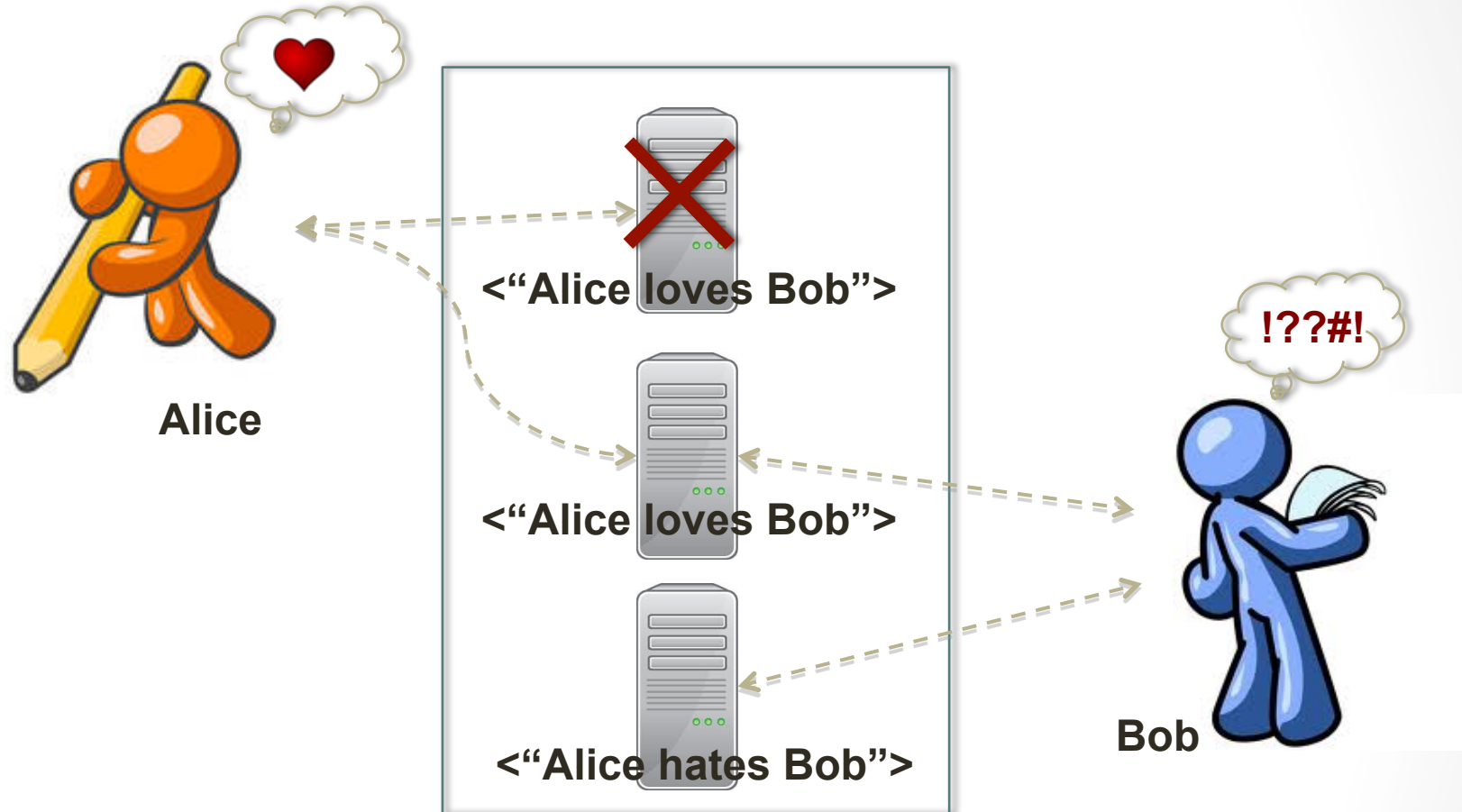
Read Protocol

- Στείλε μήνυμα `read()` σε **όλα τα αντίγραφα** και περίμενε **απάντηση από την πλειοψηφία**
- Μόλις λάβεις απάντηση επέστρεψε την τιμή που βρίσκεται μέσα στην απάντηση

Server Protocol

- Αν λάβεις μήνυμα `write(v)` άλλαξε την τιμή του αντιγράφου σου σε `v`
- Απάντησε σε όποιο μήνυμα παραλάβεις με `reply(v)` όπου `v` η τιμή του αντιγράφου σου.

Παράδειγμα Εκτέλεσης Αλγ. Β



Ανάλυση Αλγορίθμου Β

- **Δεν Ικανοποιεί Νόμιμες Ακολουθίες:** δεν μπορούμε να αναγνωρίσουμε ποια τιμή γράφτηκε τελευταία
 - Επομένως δεν ικανοποιεί καμιά Συνθήκη Συνέπειας
 - **Δεν διασφαλίζει Ατομικότητα**
- **Είναι Ανεκτικός σε σφάλματα**
 - Αφού το πολύ η μειοψηφία των αντιγράφων θα καταρρεύσει τότε πάντα υπάρχει κάποια πλειοψηφία αντιγράφων να απαντήσει σε κάθε λειτουργία
 - **Διασφαλίζει Ζωτικότητα**
- **Άρα ο Αλγόριθμος Β είναι λανθασμένος**

Αλγόριθμος Γ – Χρήση Χρονοσφραγίδας

Write Protocol

- **Αύξησε την χρονοσφραγίδα σου ($timestamp++$)**
- Στείλε μήνυμα **$write(<timestamp,v>$** σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Μόλις λάβεις απάντησεις επέστρεψε επιβεβαίωση και τερμάτισε

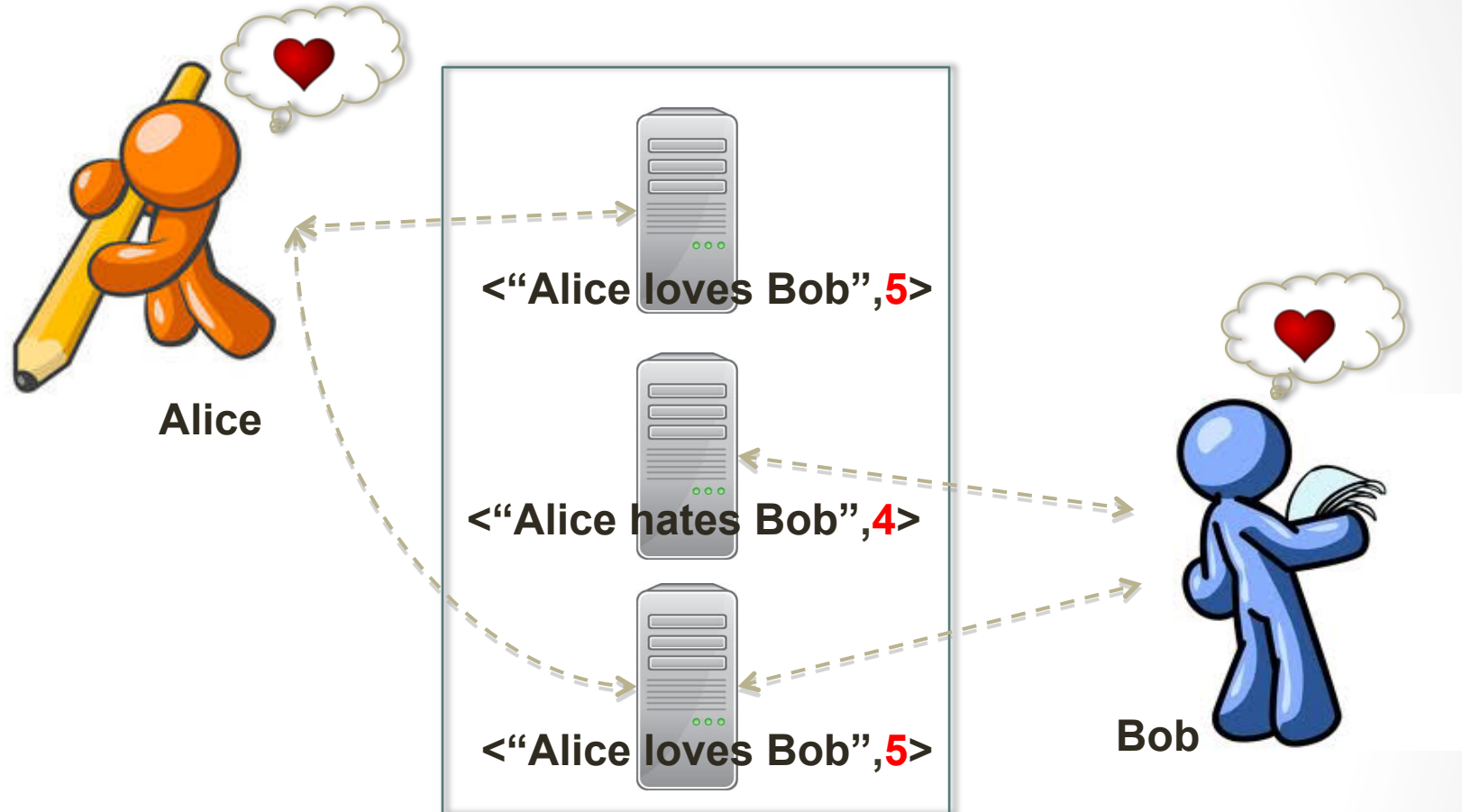
Read Protocol

- Στείλε μήνυμα $read()$ σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- **Βρες το ζεύγος $<timestamp,v>$ που περιέχει την μεγαλύτερη χρονοσφραγίδα ανάμεσα στις απαντήσεις**
- **Επέστρεψε την τιμή v που αντιστοιχεί στην μεγαλύτερη χρονοσφραγίδα**

Server Protocol

- Αν λάβεις μήνυμα **$write(<timestamp,v>$** άλλαξε την τιμή του αντιγράφου σου σε v και την **τοπική σου χρονοσφραγίδα σε $timestamp$**
- Απάντησε σε όποιο μήνυμα παραλάβεις με **$reply(<timestamp,v>$** όπου v η τιμή του αντιγράφου σου.

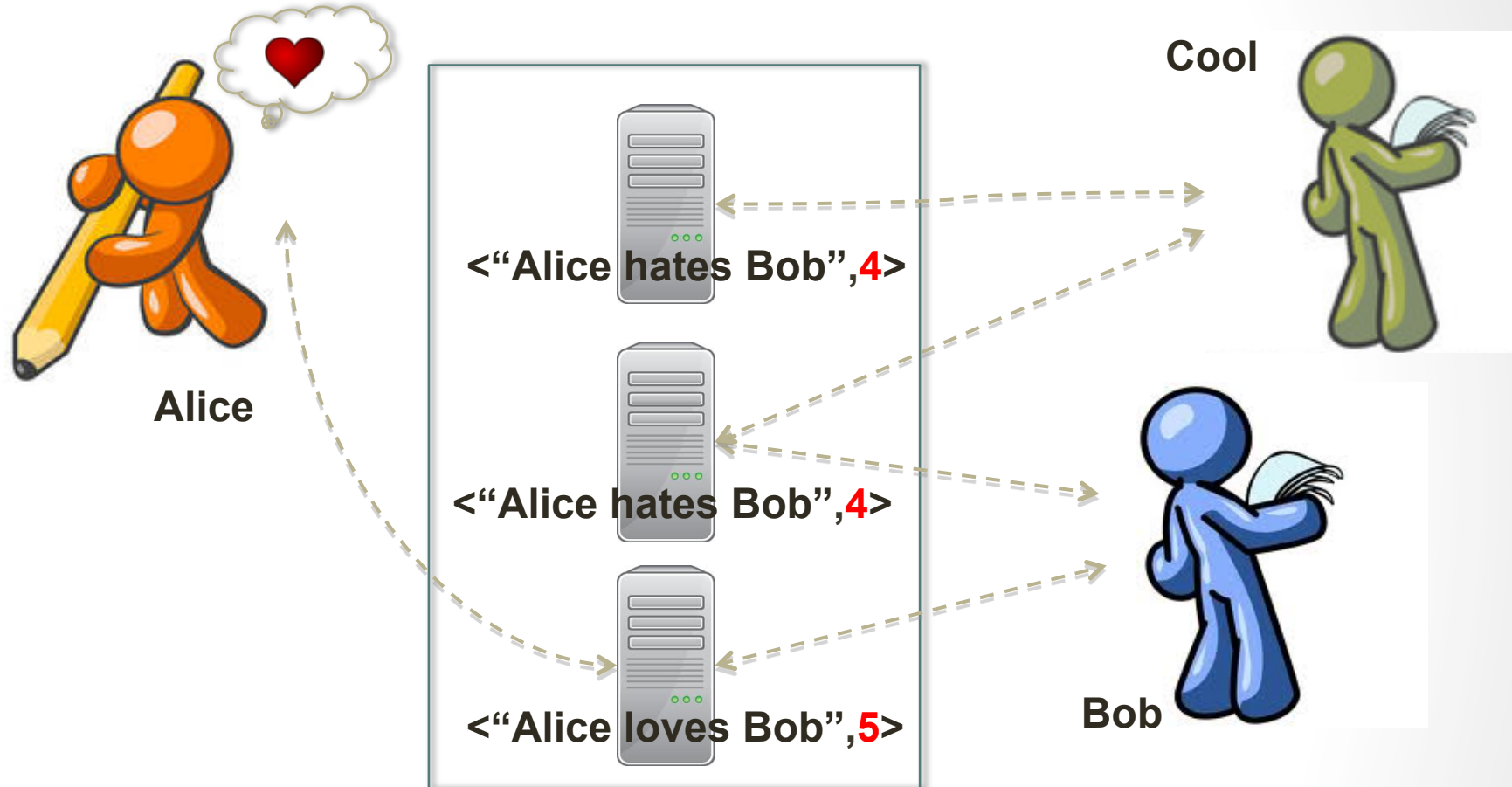
Παράδειγμα Εκτέλεσης Αλγ. Γ



Ανάλυση Αλγορίθμου Γ

- **Ικανοποιεί Νόμιμες Ακολουθίες:** κάθε ανάγνωση επιστρέφει τουλάχιστον την τιμή της τελευταίας ολοκληρωμένης εγγραφής
 - Υπάρχει **τουλάχιστον ένας εξυπηρετητής που απαντά και στις δυο λειτουργίες** ανάγνωσης και εγγραφής (τομή πλειοψηφιών)
- **Είναι Ανεκτικός σε σφάλματα**
 - Αφού το πολύ η μειοψηφία των αντιγράφων θα καταρρεύσει τότε πάντα υπάρχει κάποια πλειοψηφία αντιγράφων να απαντήσει σε κάθε λειτουργία
 - **Διασφαλίζει Ζωτικότητα**
- **Ο Αλγόριθμος Γ ικανοποιεί την Ατομικότητα?**

Αλγ. Γ με πολλούς Αναγνώστες



Πρόβλημα Αλγόριθμου Γ

- Δεν Ικανοποιεί την χρονική σειρά των λειτουργιών
 - Δεν διασφαλίζει Ατομικότητα
- Διασφαλίζει Κανονικότητα

Αλγόριθμος ABD

Write Protocol

- Αύξησε την χρονοσφραγίδα σου ($\text{timestamp}++$)
- Στείλε μήνυμα $\text{write}(\langle \text{timestamp}, v \rangle)$ σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Μόλις λάβεις απάντησεις επέστρεψε επιβεβαίωση και τερμάτισε

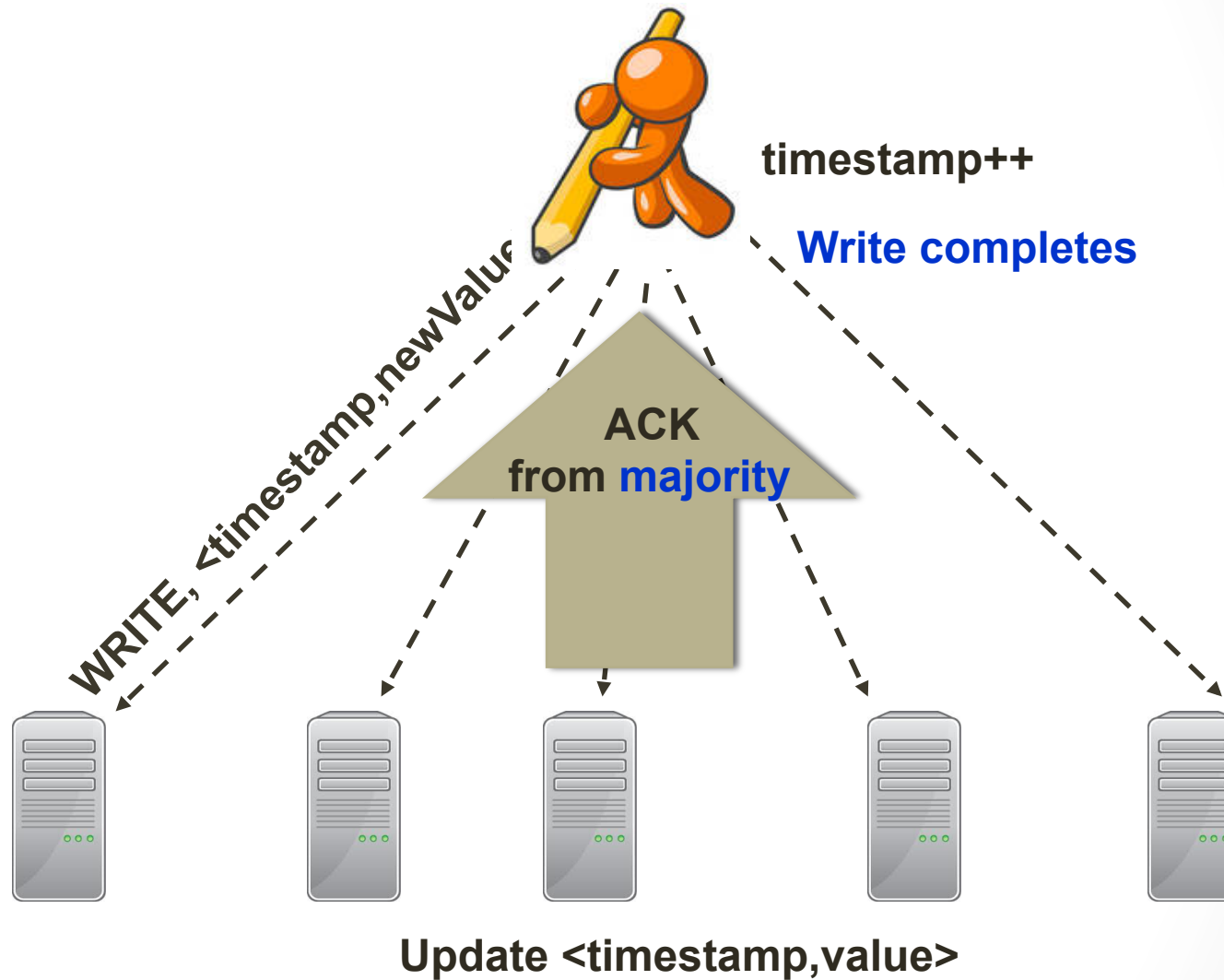
Read Protocol

- Στείλε μήνυμα $\text{read}()$ σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Βρες το ζεύγος $\langle \text{timestamp}, v \rangle$ που περιέχει την μεγαλύτερη χρονοσφραγίδα ανάμεσα στις απαντήσεις
- Στείλε μήνυμα $\text{write}(\langle \text{timestamp}, v \rangle)$ σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Επέστρεψε την τιμή v που αντιστοιχεί στην μεγαλύτερη χρονοσφραγίδα

Server Protocol

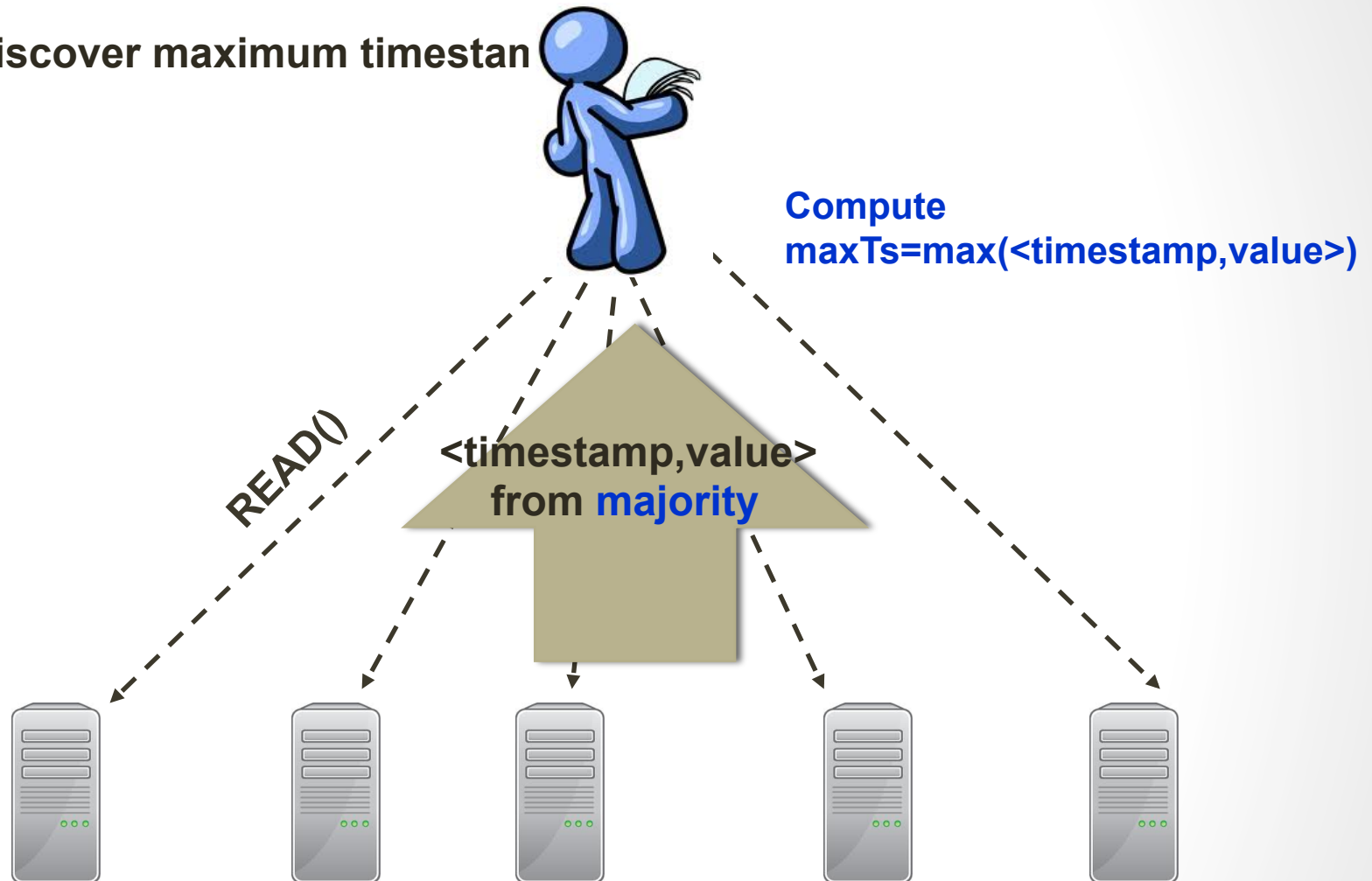
- Αν λάβεις μήνυμα $\text{write}(\langle \text{timestamp}, v \rangle)$ και $\text{local.timestamp} < \text{msg.timestamp}$ άλλαξε την τιμή του αντιγράφου σου σε v και την τοπική σου χρονοσφραγίδα σε timestamp
- Απάντησε σε όποιο μήνυμα παραλάβεις με $\text{reply}(\langle \text{timestamp}, v \rangle)$ όπου v η τιμή του αντιγράφου σου.

ABD: Εγγραφείας



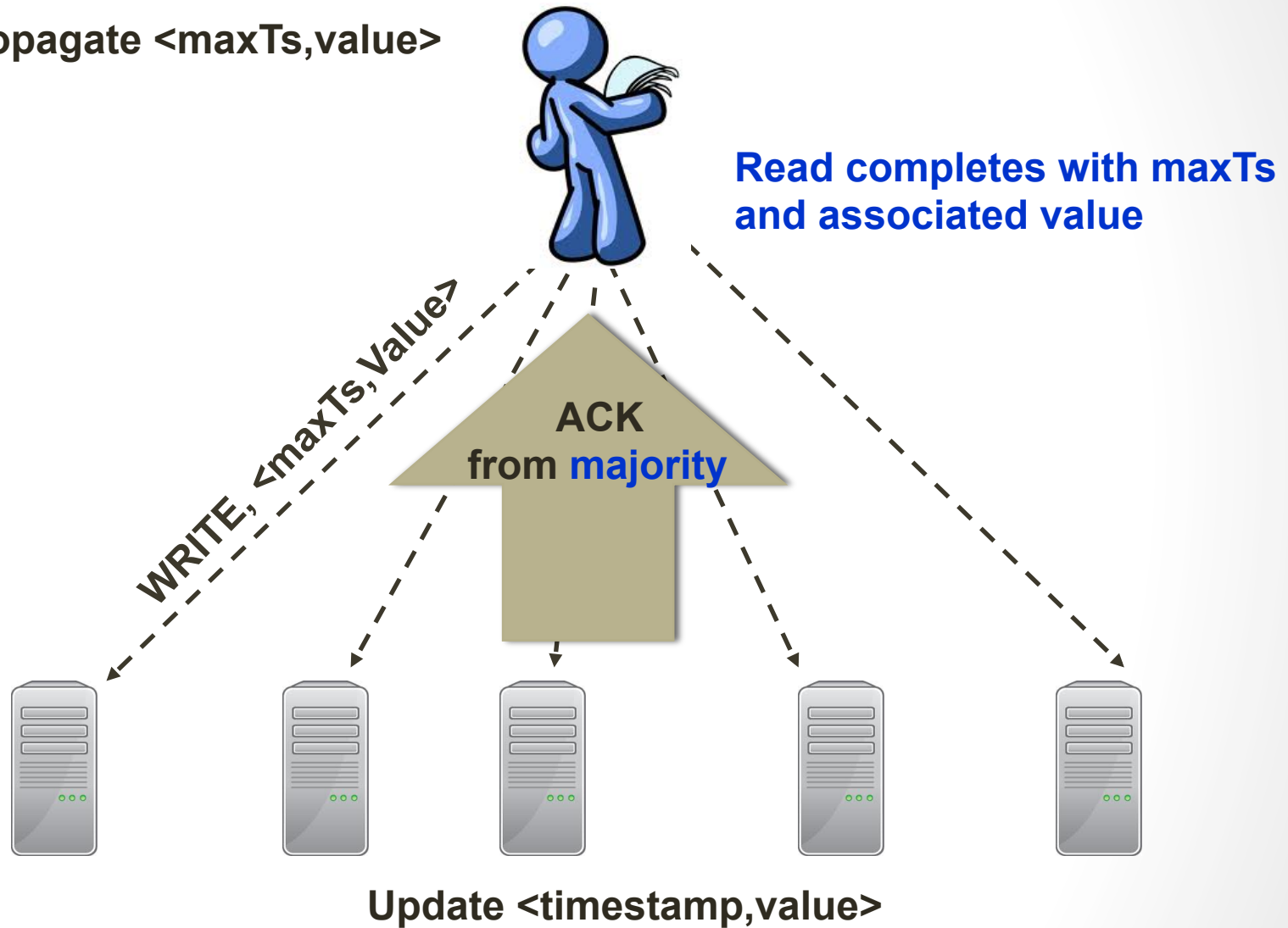
ABD: Αναγνώστης

Phase1: Discover maximum timestamp

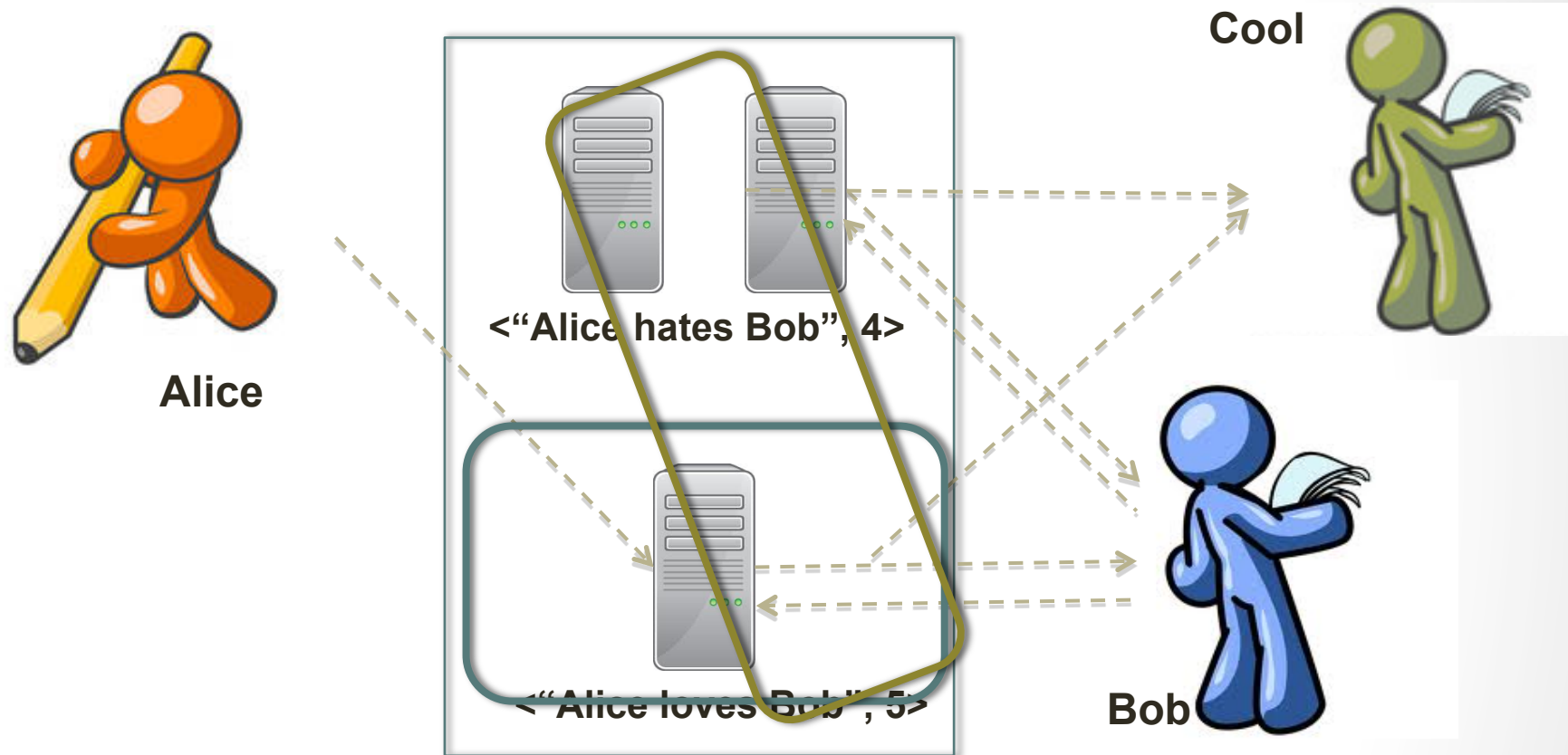


ABD: Αναγνώστης

Phase2: Propagate $\langle \text{maxTs}, \text{value} \rangle$



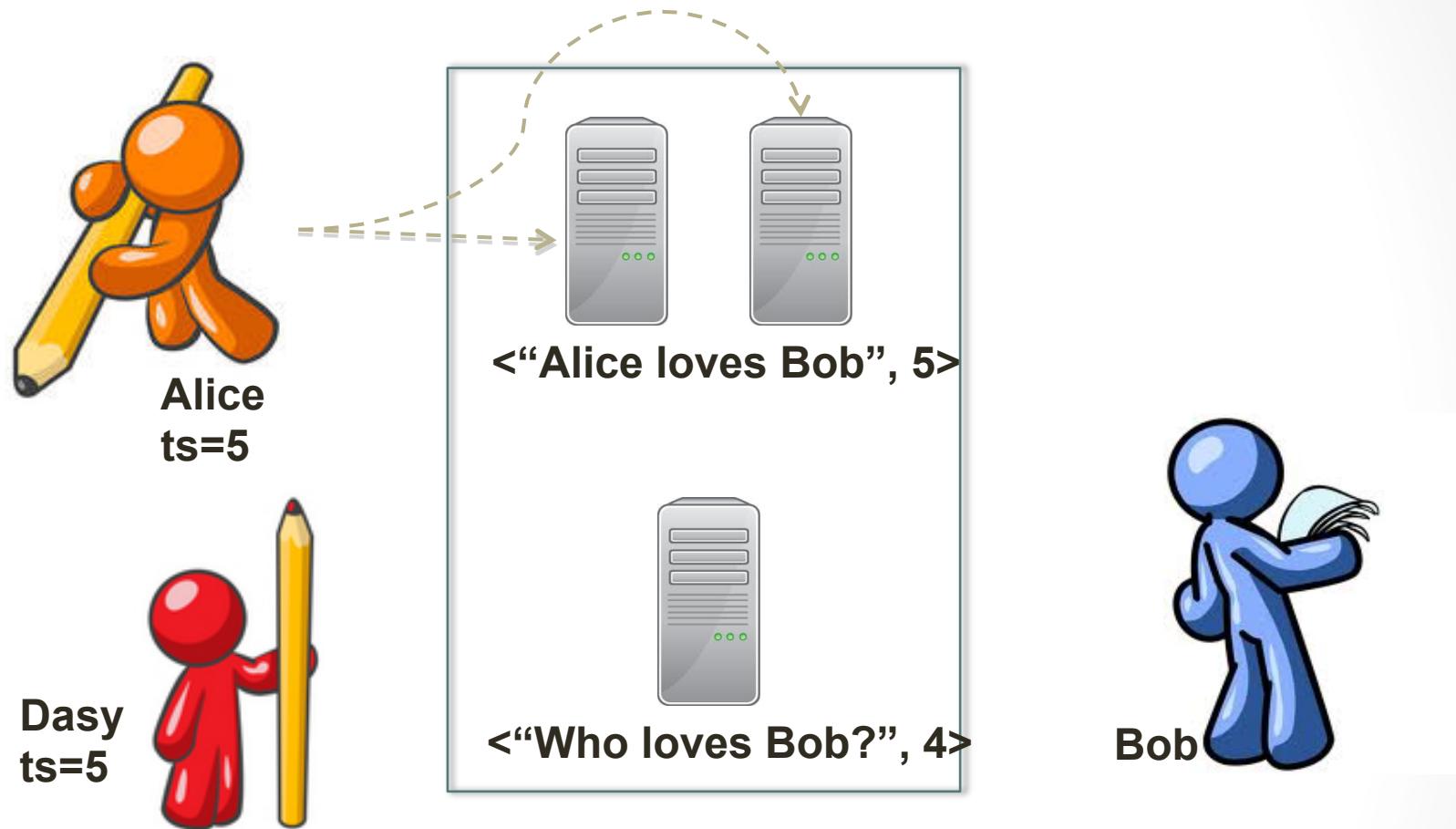
Παράδειγμα Εκτέλεσης ABD



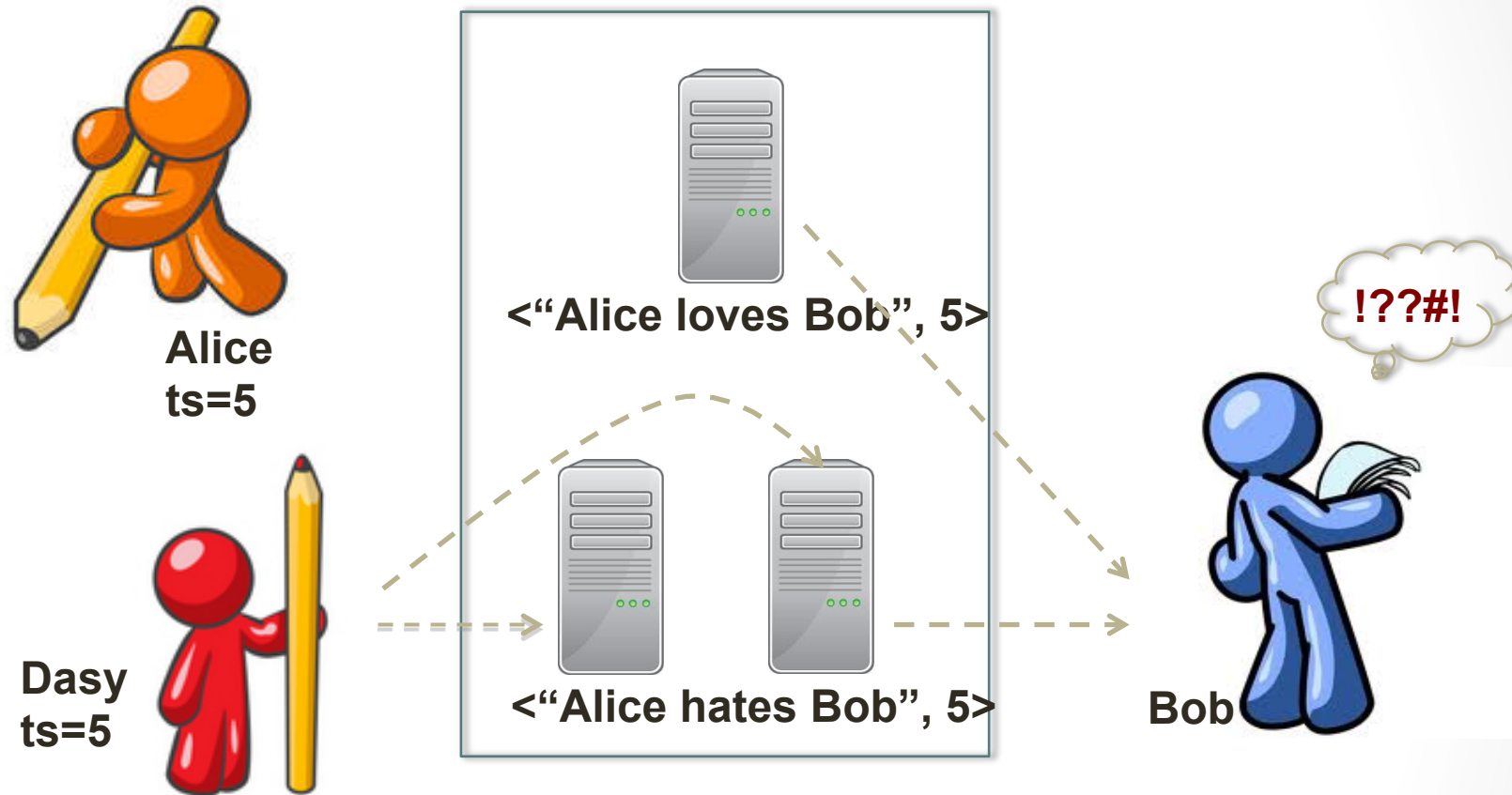
Ανάλυση Αλγορίθμου ABD

- **Ικανοποιεί Ατομικότητα**
 - **Ικανοποιεί Νόμιμες Ακολουθίες:** Όπως Αλγόριθμος Γ
 - **Ικανοποιεί Χρονική Σειρά λειτουργιών:** Πριν ολοκληρωθεί κάποια ανάγνωση η τιμή που επιστρέφεται στέλνεται στην πλειοψηφία των αντιγράφων
 - Επόμενες αναγνώσεις θα δουν την τιμή αυτή σε τουλάχιστον ένα αντίγραφο
- **Διασφαλίζει Ζωτικότητα**
 - **Είναι Ανεκτικός σε σφάλματα:** Αφού το πολύ η μειοψηφία των αντιγράφων θα καταρρεύσει τότε πάντα υπάρχει κάποια πλειοψηφία αντιγράφων να απαντήσει σε κάθε λειτουργία
- **Ο Αλγόριθμος ABD υλοποιεί Ατομική ΚΚΜ Εγγραφής/Ανάγνωσης στο ασύγχρονο, SWMR μοντέλο**

Τι συμβαίνει στο MWMR μοντέλο



Τι συμβαίνει στο MWMR μοντέλο



Πως επιλύουμε το πρόβλημα;

- Ο εγγραφέας πρέπει να ανακαλύψει την **τελευταία χρονοσφραγίδα** πριν την αυξήσει
- Κάθε εγγραφέας περιλαμβάνει σε κάθε μήνυμα τον AT του
 - Σπάζουμε την συμμετρία σε περίπτωση που δύο εγγραφείς προσπαθήσουν να χρησιμοποιήσουν την ίδια χρονοσφραγίδα
- Λεξικογραφική σύγκριση Χρονοσφραγίδων

Γενικοποίηση Αλγόριθμου ABD στο MWMR

Write Protocol

- Φάση 1: Στείλε μήνυμα `read()` σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Φάση 2:
 - Βρες το `<timestamp,id,v>` που περιέχει το μεγαλύτερη χρονοσφραγίδα ανάμεσα στις απαντήσεις
 - Αύξησε την χρονοσφραγίδα (`timestamp++`) και στείλε μήνυμα `write(<timestamp,ID,v>)` σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Μόλις λάβεις απαντήσεις επέστρεψε επιβεβαίωση και τερμάτισε

Read Protocol

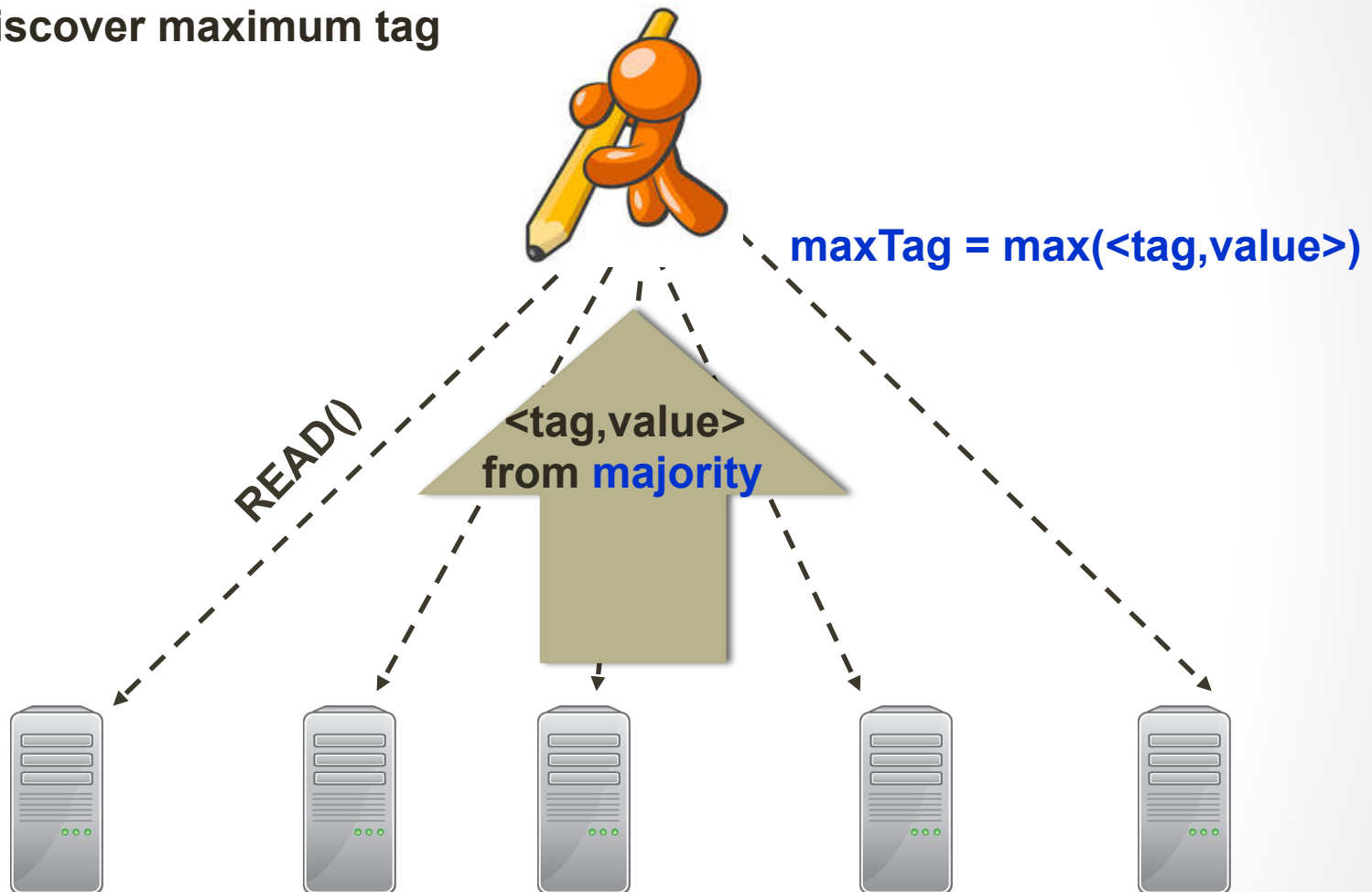
- Φάση 1: Στείλε μήνυμα `read()` σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Φάση 2:
 - Βρες το `<timestamp,id,v>` που περιέχει την μεγαλύτερη χρονοσφραγίδα ανάμεσα στις απαντήσεις
 - Στείλε μήνυμα `write(<timestamp,id,v>)` σε όλα τα αντίγραφα και περίμενε απάντηση από την πλειοψηφία
- Επέστρεψε την τιμή `v` που αντιστοιχεί στην μεγαλύτερη χρονοσφραγίδα

Server Protocol

- Αν λάβεις μήνυμα `write(<timestamp,id,v>)` και `local.timestamp < msg.timestamp` άλλαξε την τιμή του αντιγράφου σου σε `v` και την τοπική σου χρονοσφραγίδα σε `<timestamp,id>`
- Απάντησε σε όποιο μήνυμα παραλάβεις με `reply(<timestamp,id,v>)` όπου `v` η τιμή του αντιγράφου σου.

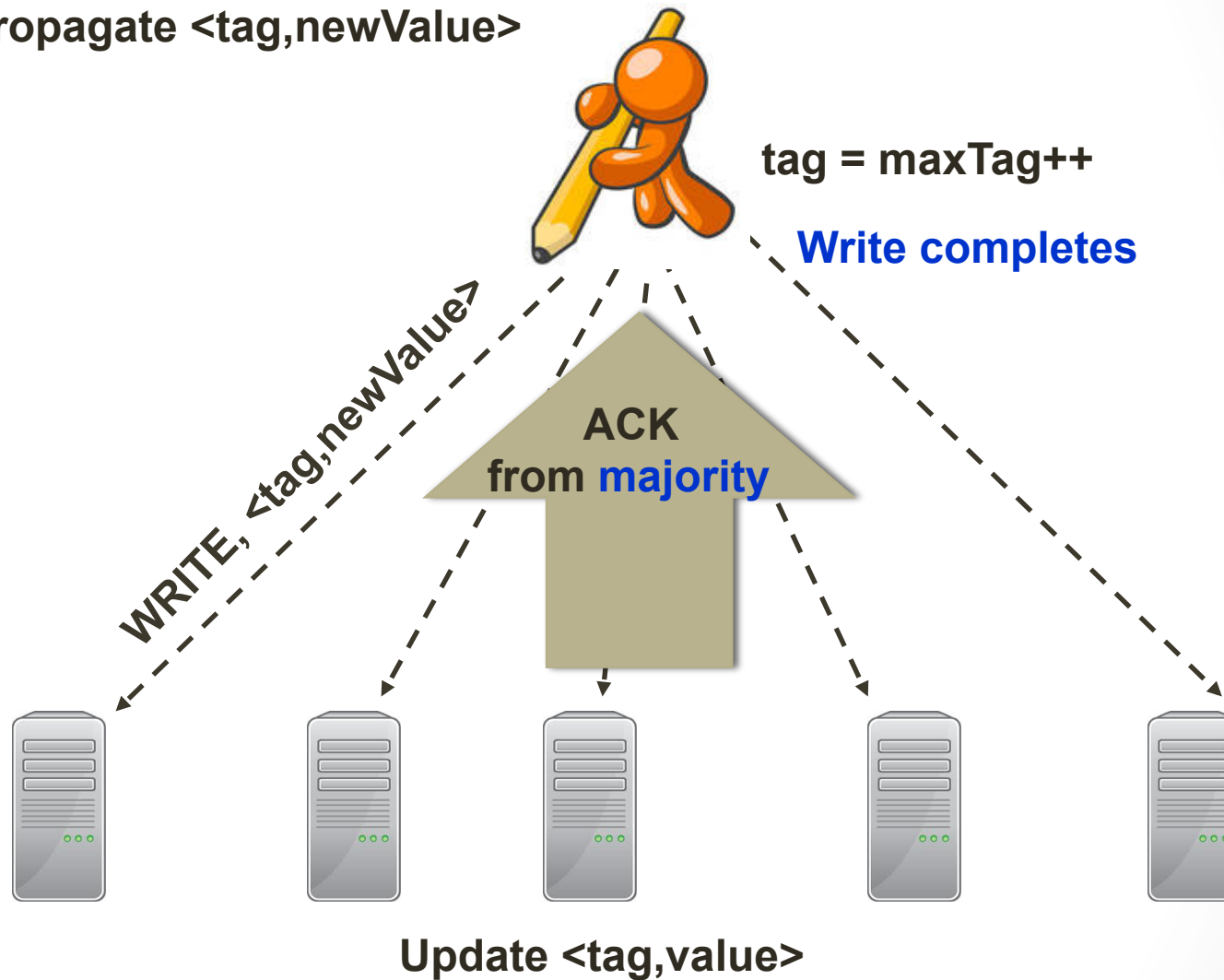
Εγγραφές 2 Φάσεων

Phase1: Discover maximum tag

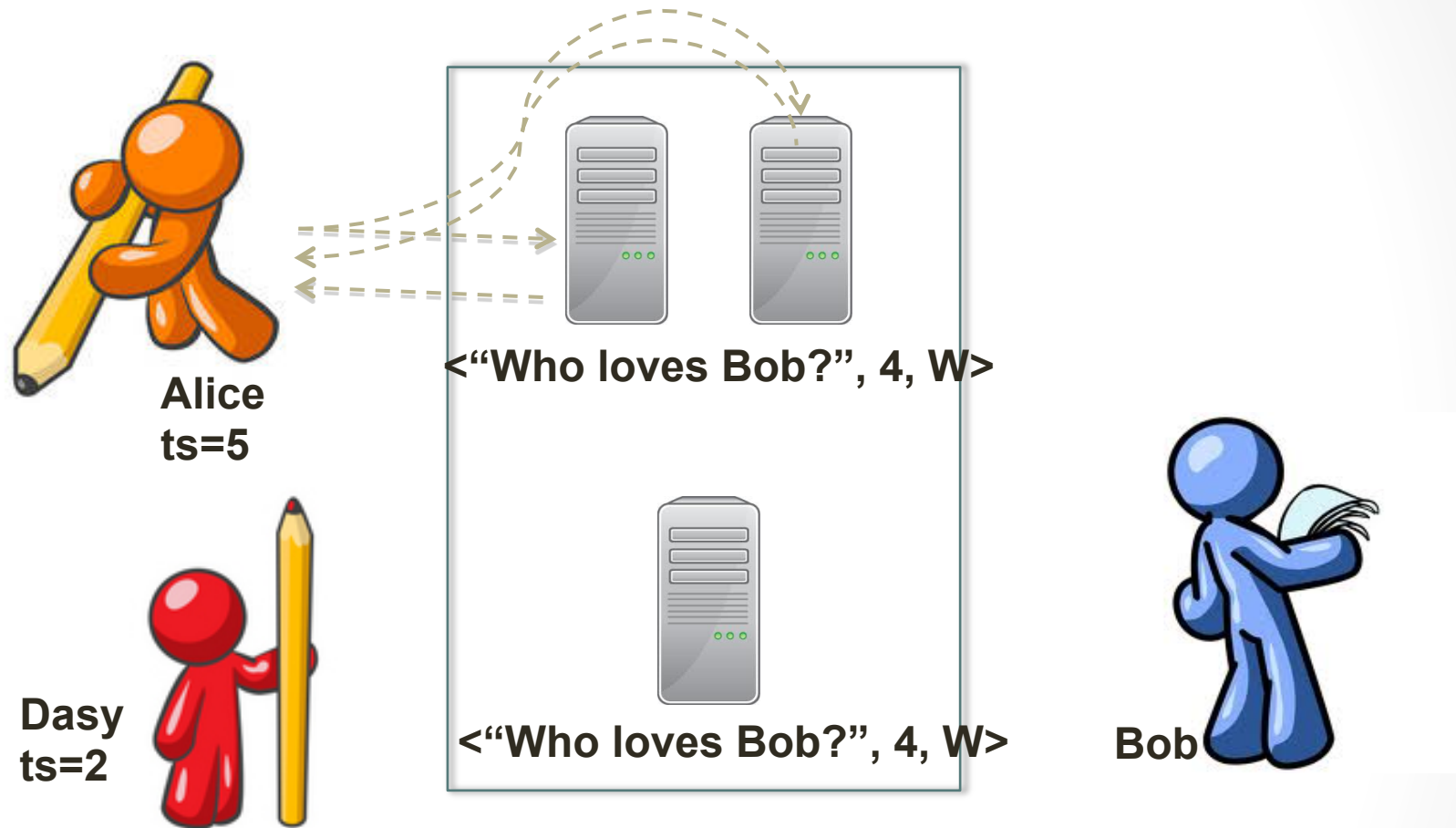


Εγγραφές 2 Φάσεων

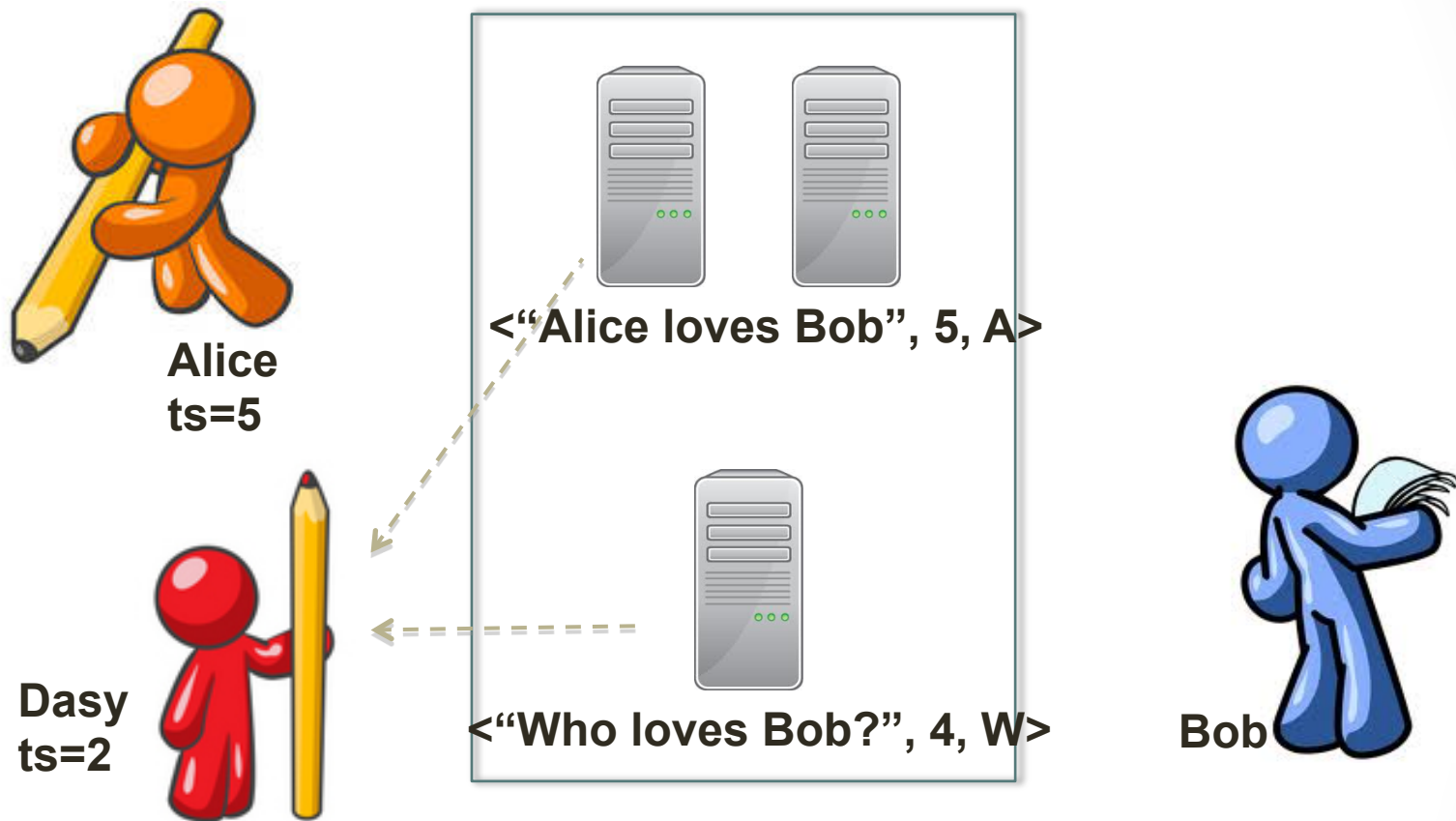
Phase2: Propagate $\langle \text{tag}, \text{newValue} \rangle$



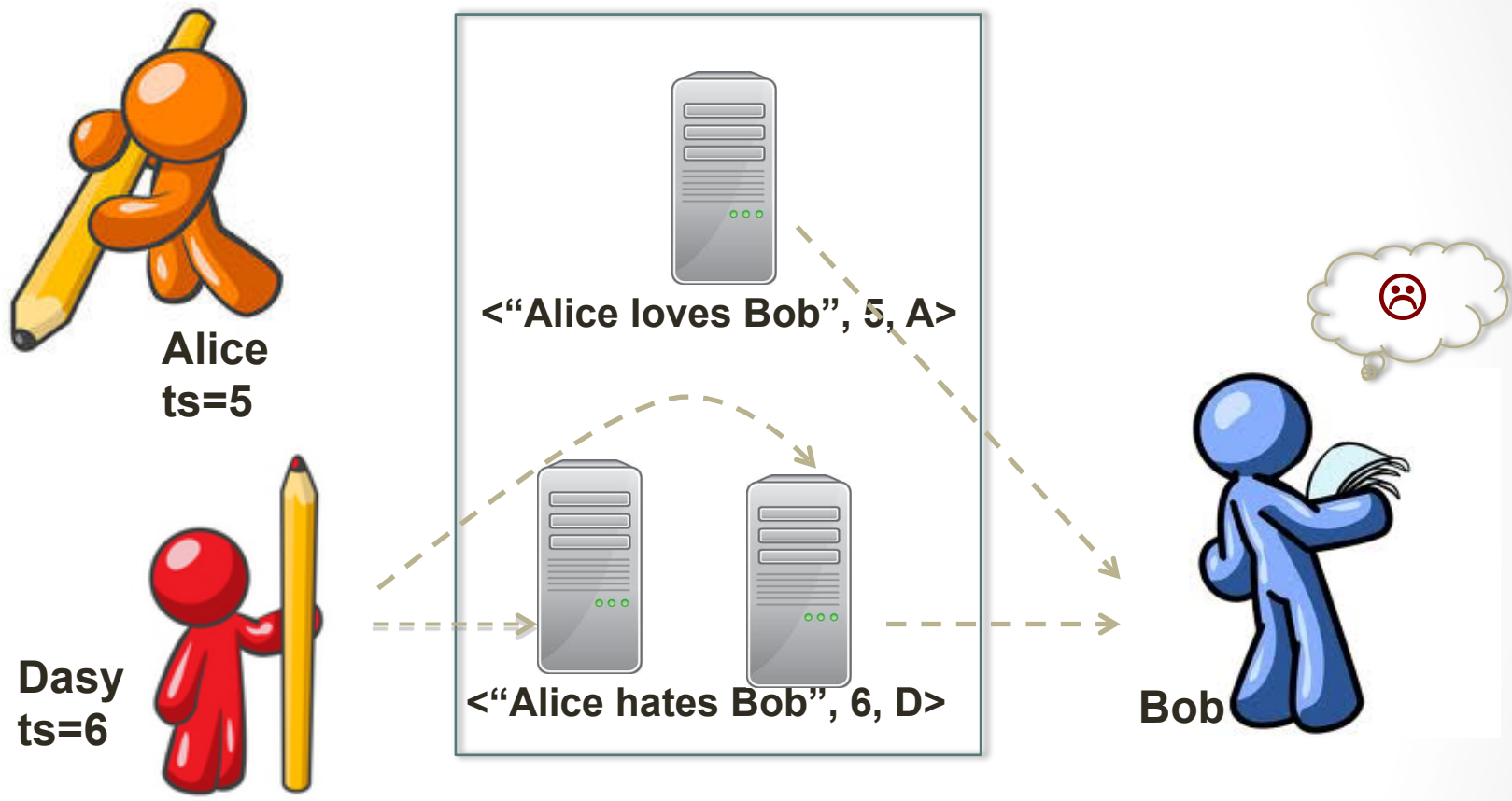
Παράδειγμα Εκτέλεσης



Παράδειγμα Εκτέλεσης



Παράδειγμα Εκτέλεσης



Ανάλυση Ορθότητας

- **Ζωτικότητα: Ικανοποιείται**
 - Πάντα υπάρχει πλειοψηφία εξυπηρετητών να απαντήσει σε μια λειτουργία
- **Ατομικότητα: Ικανοποιείται**
 - Αναγνώσεις επιστρέφουν τουλάχιστον τελευταία εγγραφή
 - Αν μια εγγραφή έπεται μιας άλλης εγγραφής τότε διασφαλίζει μεγαλύτερη χρονοσφραγίδα
 - Αν μια ανάγνωση έπεται μια άλλης ανάγνωσης τότε επιστρέφει νεότερη ή τουλάχιστον την ίδια τιμή

Ερωτήσεις;

