

# Πρότυπα Ανάπτυξης Λογισμικού

Παρουσίαση Προτύπων:

Circuit Breaker

Competing Consumers



# Circuit Breaker Pattern

# Ποιο είναι το πρόβλημα;

Είναι σύνηθες για τα λειτουργικά συστήματα να κάνουν κλήσεις σε άλλα συστήματα που τρέχουν σε διαφορετικές διεργασίες, πολλές φορές σε απομακρυσμένες μηχανές που βρίσκονται σε άλλο δίκτυο.

Η βασική διαφορά που υπάρχει ανάμεσα στην κλήση διεργασιών από τη μνήμη και στην κλήση από ένα απομακρυσμένο σύστημα είναι ότι στη δεύτερη περίπτωση η κλήση μπορεί να αποτύχει ή να μην απαντηθεί.

Αυτό το πρόβλημα μπορεί να αποδειχτεί κρίσιμο όταν υπάρχουν πολλές διεργασίες οι οποίες στέλνουν αιτήματα σε μια απομακρυσμένη διεργασία που δεν αποκρίνεται, δημιουργώντας έτσι πολλά προβλήματα μέσα στο πρόγραμμα.

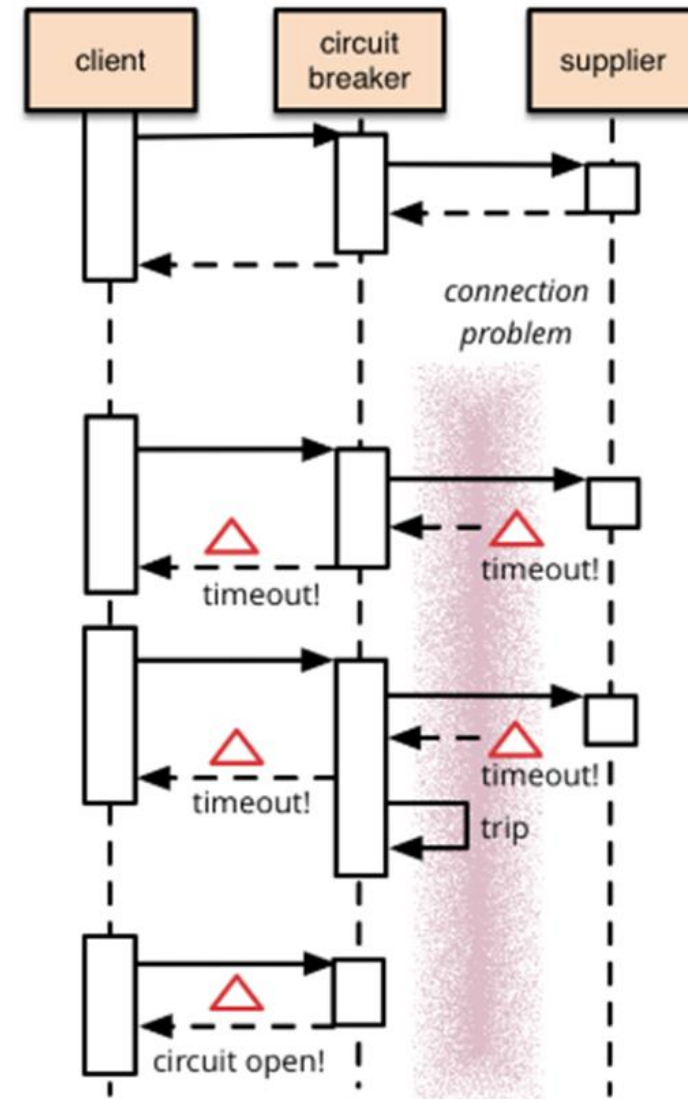
## Τι είναι το Circuit Breaker;

- ▶ Είναι μια διεργασία η οποία χρησιμοποιείται σαν μεσάζοντας για την επικοινωνία μεταξύ διεργασιών.
- ▶ Όπως λέει και το όνομα είναι ένας διακόπτης.

# Πως λειτουργεί το Circuit Breaker;

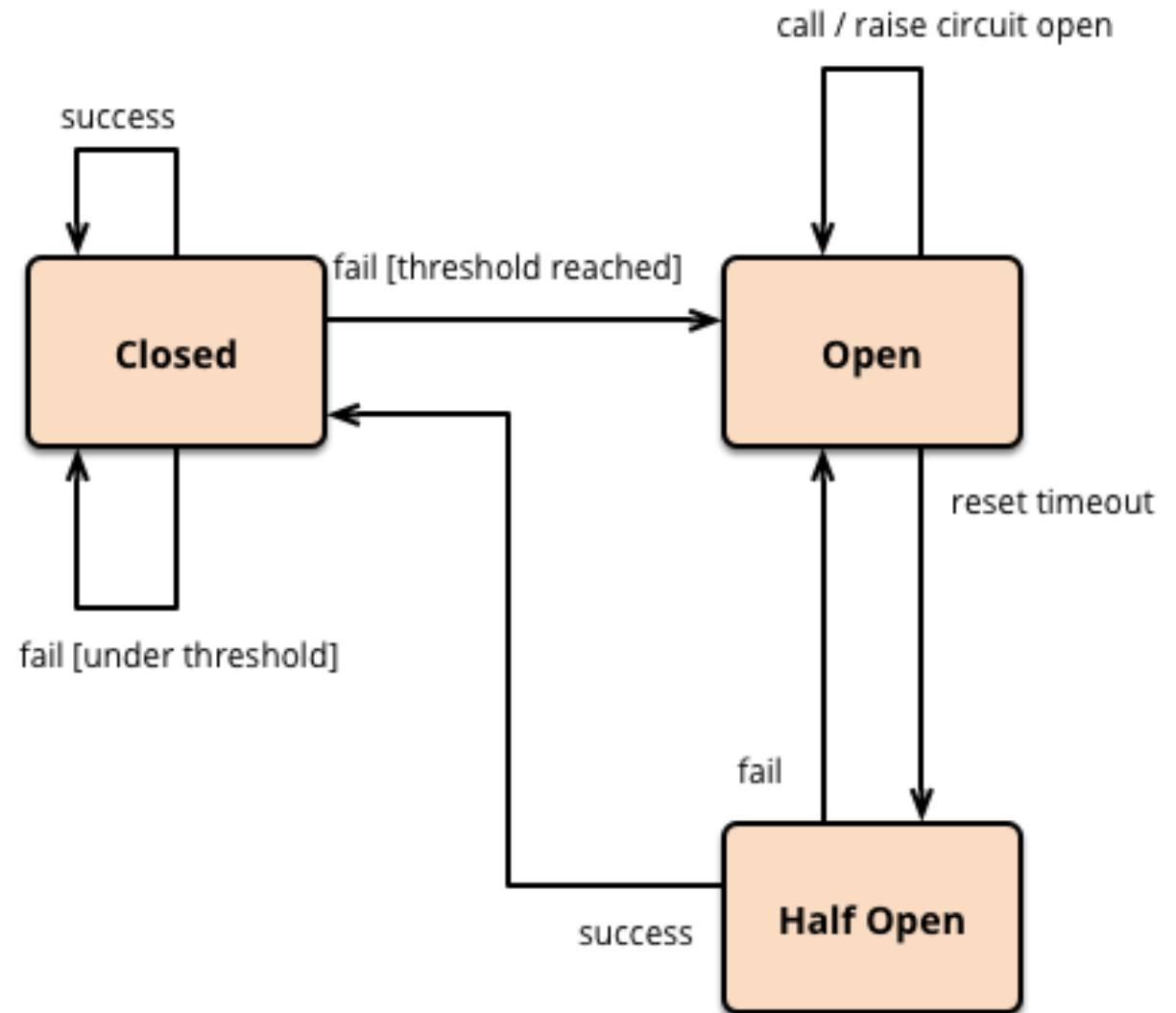
Το circuit breaker έχει κάποια ορίσματα κάποια εκ των οποίων είναι το όριο αποτυχιών (Failure Threshold) και ο μετρητής (Counter). Ο μετρητής μετράει τον αριθμό των φορές που δεν κατάφερε να υπάρξει σύνδεση και όταν φτάσει το failure threshold τότε το circuit breaker ενεργοποιείται. Αν δεν υπάρχει πρόβλημα σύνδεσης ο μετρητής μηδενίζεται.

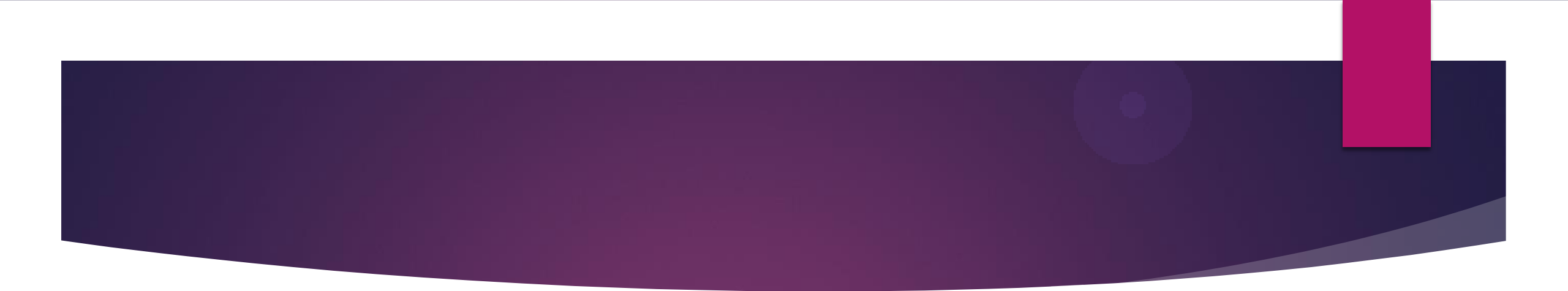
Όσο βρίσκεται στην κατάσταση «Ανοικτό» επιστρέφει μήνυμα σφάλματος χωρίς να επικοινωνήσει με την μη ανταποκρίσιμη εφαρμογή (όπως φαίνεται και στο διπλανό σχήμα).



# Πως ξανακλείνει το κύκλωμα ;

- ▶ Όταν το κύκλωμα είναι ανοικτό οι διεργασίες δεν επικοινωνούν με την απομακρυσμένη διεργασία.
- ▶ Για να ξανακλείσει το κύκλωμα θα πρέπει να γίνει μια δοκιμή σύνδεσης αν πετύχει η σύνδεση γίνεται άρση του αποκλεισμού.



- 
- ▶ Αυτό μπορεί να επιτευχθεί με τη χρήση μιας τρίτης κατάστασης “Half-Open”. Σε αυτή την κατάσταση το κύκλωμα αφήνει ορισμένα αιτήματα να περάσουν. Αν η διαδικασία πραγματοποιηθεί χωρίς κάποιο σφάλμα τότε το κύκλωμα μεταβαίνει στην κατάσταση «Κλειστό», αν αντιθέτως υπάρξει κάποιο πρόβλημα το κύκλωμα μπαίνει στην κατάσταση «Ανοικτό»



# Competing Consumers Pattern



## Ένα συνηθισμένο πρόβλημα...

Ένα πρόβλημα που παρουσιάζεται συνήθως σε εφαρμογές cloud είναι ο κατακλυσμός από εκατοντάδες μηνύματα.

Αυτό έχει ως συνέπεια:

- Απρόβλεπτο φόρτο εργασίας στις ώρες αιχμής.
- Μπλοκάρισμα εφαρμογής.
- Ή παραποίηση μηνυμάτων.

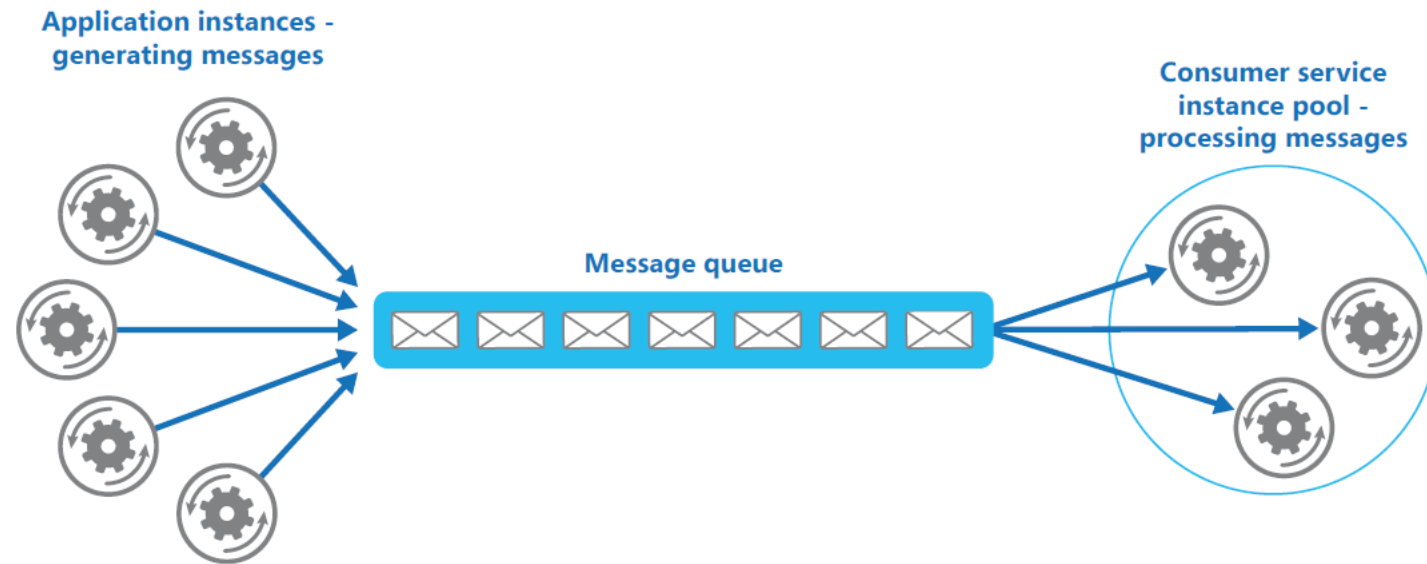
# Το πρότυπο Competing Consumers

Μια λύση που προτείνεται είναι η χρήση του μοντέλου Competing Consumers.

Μια σύντομη επεξήγηση του μοντέλου:

- Χρήση ουράς για την αποθήκευση των μηνυμάτων
- Διαχείριση αιτημάτων από οποιοδήποτε instance

# Δομή του προτύπου Competing Consumers



# Πλεονεκτήματα

1. Βελτιώνει την αξιοπιστία αποτρέποντας την επικοινωνία χρήστη – συστήματος
2. Είναι μεταβαλλόμενο σε μέγεθος εργασίας
3. Δεν χρειάζεται περίπλοκο συντονισμό
4. Βελτίωση ανθεκτικότητας σε περίπτωση ύπαρξης συναλλαγών
5. Λειτουργεί ως ρυθμιστής του όγκου διαχείρισης

# Επιπλέον Καταστάσεις προς Υλοποίηση

1. Σειρά Μηνυμάτων
2. Ανεξαρτησία Λειτουργιών – Ανθεκτικότητα
3. Διαχείριση Αποτελεσμάτων
4. Εύρεση Κακόβουλων Μηνυμάτων
5. Μεταβολή Όγκου Συστήματος Παροχής Μηνυμάτων
6. Εξασφάλιση Αξιοπιστίας

## Χρήση του προτύπου όταν...

1. Το φόρτο εργασίας διαιρείται σε ασύγχρονες εργασίες
2. Οι εργασίες είναι ανεξάρτητες και τρέχουν παράλληλα
3. Διαθεσιμότητα / ανθεκτικότητα λύσης σε τυχόν καταστροφή εργασίας

# Αποφυγή χρήσης όταν...

1. Αδυναμία διάσπασης όγκου εργασίας
2. Ταυτόχρονη εκτέλεση εργασιών
3. Συγκεκριμένη σειρά εκτέλεσης