

# Κατανεμημένα συστήματα

## Άσκηση 1

### Goal 1: connecting users and resources

What are the types of resources shared? How does web address the two problems of sharing?

Το web πληροί αυτόν τον στόχο αφού μέσω αυτού διαμοιράζονται αρκετοί πόροι.

- Αρχεία (κείμενο εικόνες ...)
- Υπολογιστές. Ένα σύνολο από servers εξυπηρετεί κάποιον client στο web.
- Εκτυπωτές. Είναι δυνατόν κάποιος να έχει shared τον εκτυπωτή του μέσω του web.
- Storage facilities...

Τα δύο βασικά προβλήματα αντιμετωπίζονται

Security

1. Encryption
2. Authentication (digital signatures)

Unwanted communication

1. Password protection
2. Authentication. (παράδειγμα να επιτραπεί η χρήση σε συγκεκριμένες ip, cookies κτλ...)
3. Filters (π.χ. firewalls)

### Goal 2: Transparency

For each type of transparency, explain the transparency degree offered by the web.

#### 1. Access transparency

- Υπάρχει πλήρης Access transparency. Ο χρήστης απλά πληκτρολογεί ένα URL και με τρόπο απόλυτα κρυμμένο από αυτόν τα δεδομένα εμφανίζονται στην οθόνη του. Ο τρόπος αναπαράστασης των δεδομένων αυτών δεν τον απασχολεί αφού τα «κατώτερα» επίπεδα φροντίζουν να τα διαχειριστούν χωρίς την παρέμβασή του. Ο χρήστης δεν γνωρίζει αν ο υπολογιστής που συνδέεται είναι SUN/PC/SG/MAC κτλ (εκτός και αν το ψάξει). Ούτε τον ενδιαφέρει τι λειτουργικό σύστημα τρέχει κτλ...
- Επίσης η αποδοχή από όλους τους κατασκευαστές μηχανημάτων και λειτουργικών συστημάτων, κάποιων προτύπων (RFC) έχει συμβάλει στην πλήρη επιτυχία της access transparency. Δηλαδή έχουν επιλυθεί όλα τα προβλήματα συμβατότητας (π.χ. στο TCP/IP όλα τα

δεδομένα αποστέλλονται σε big endian). Έτσι στον κοινό χρήστη όλα τα “δεδομένα” φαίνονται με την ίδια μορφή.

2. Location transparency
  - Θα μπορούσε να χαρακτηριστεί ως πλήρης location transparency. Βλέποντας κάποιος ένα URL δεν μπορεί να βγάλει κανένα συμπέρασμα από την θέση μιας resource. Επιπλέον μπορεί κάποιος να χρησιμοποιήσει ένα resource χωρίς να ξέρει που είναι. Πιθανώς στο URL να φανεί η χώρα. Σε γενικές γραμμές πάντως το URL δεν περιέχει σημαντικές πληροφορίες για την περιοχή ενός resource. Δηλαδή υπάρχουν μηχανισμοί οι οποίοι μας εξασφαλίζουν την εύκολη και γρήγορη αναζήτηση της πηγής των δεδομένων χωρίς να γνωρίζουμε εκ των προτέρων τίποτε περισσότερο από το URL του.
3. Migration transparency.
  - Φυσικά μπορεί κάποια resource να μετακινηθεί σε μια άλλη περιοχή χωρίς αυτό να γίνει αντιληπτό (τουλάχιστον στον απλό χρήστη). Αρκεί να γίνει σωστή ενημέρωση του DNS server ώστε όταν του ζητείται αυτό το ip-name να δίνει την νέα ip. Η αλλιώς μπορεί να υπάρχει κάποιο μηχάνημα που να κάνει redirect.
4. Relocation transparency.
  - Το web μπορεί να έχει «από κάτω» οποιαδήποτε είδους δίκτυα. Ακόμα και ασύρματα / κινητά. Μπορεί λοιπόν κάποιο resource να μετακινηθεί καθώς χρησιμοποιείται.
5. Replication transparency
  - Μπορούμε να έχουμε replicas με το ίδιο όνομα. Κάθε φορά που γίνεται μια σύνδεση υπάρχει ένα front-end που μας συνδέει στο καταλληλότερο. (π.χ. server clusters, mirroring, content delivery network)
6. Concurrency transparency
  - Όταν κάποιος χρήστης συνδέεται σε μια υπηρεσία του Web δεν μπορεί να γνωρίζει πόσοι οι ποιοι χρήστες είναι συνδεδεμένοι σε αυτόν. Επιπλέον μπορούν οι εφαρμογές στο Web που επιδέχονται αλλαγές από τους χρήστες να μπορούν να εξασφαλίσουν την ακεραιότητα των δεδομένων ακόμη και στις περιπτώσεις που πολλοί χρήστες ταυτόχρονα προσπαθούν να τροποποιήσουν κοινά δεδομένα.
7. failure transparency.
  - Χωρίς να το γνωρίζουμε μπορούν να συμβαίνουν αποτυχίες και να επανακάμπτει το σύστημα στο οποίο συνδεόμαστε. Για παράδειγμα μπορούμε κάτω από έναν web server να έχουμε ένα cluster από servers εκ των οποίων ο ένας να πέσει αλλά εμείς να μην το καταλάβουμε. Επιπλέον το DNS επιστρέφει πολλαπλά ονόματα για κάποια αναζήτηση ενός ονόματος. Υφίσταται το πρόβλημα του να εντοπίσουμε αν ένα resource έπεσε ή απλά αργεί.
8. Persistence transparency
  - Δεν μπορούμε να γνωρίζουμε τέτοιου είδους πληροφορίες. Άρα υπάρχει persistence transparency.

## 9. Performance transparency

- Μπορεί το σύστημα να κάνει αλλαγές ώστε να κάνει load balancing χωρίς αυτό να γίνει αντιληπτό από τον χρήστη του web.

## 10. Scale transparency

- Μπορεί το web να γίνει scaled τόσο σε μέγεθος όσο και σε επίπεδο εφαρμογών / πρωτοκόλλων χωρίς να αλλάζει δραστικά η δομή του (π.χ. προσθήκη νέων http εντολών ή νέου domain στο δίκτυο, σύνδεση νέων χρηστών κτλ...)

### **Goal 3: Openness**

To what extent is web an open system? Briefly mention the protocols used by the web. Give an example of a web policy and a web mechanism (other than caching).

Το web είναι open σύστημα. Σε αυτό μπορούν να συνδεθούν διαφορετικά συστήματα και χρήστες χρησιμοποιώντας κοινά πρωτόκολλα διασύνδεσης τα οποία είναι καλά ορισμένα. Επιπλέον στο web μπορούν να συνδεθούν διαφορετικών ειδών συστήματα χρησιμοποιώντας το ίδιο interface (portability).

Τα σημαντικότερα πρωτόκολλα είναι:

- http, ftp, gopher (!!!) , telnet
- dns (naming services)
- smtp (e-mail)
- tcp/ip (TCP – UDP connections)
- Security protocols (SSL, TLS)
- Authentication protocols
- Και πάρα πολλά άλλα...(ssh, ...)

#### Web policy/mechanism

Ένα παράδειγμα mechanism είναι ότι το web υποστηρίζει το τρέξιμο κώδικα που κατεβαίνει μαζί με μία ιστοσελίδα (javascript). Το τι θα μπορεί να κάνει αυτός ο κώδικας (το πόσα πολλά πράγματα θα μπορεί να κάνει ή το τι περιορισμούς θα έχει) το καθορίζει ο χρήστης με policy. Για παράδειγμα κάποιος μπορεί να μην θέλει να εκτελείται καθόλου τέτοιος κώδικας, ενώ κάποιος άλλος να το επιτρέπει αρκεί να μην προκαλεί ζημιά στο σύστημα.

Ένα άλλο παράδειγμα είναι το security. Το web προσφέρει μηχανισμούς για ασφαλή επικοινωνία. Το αν θα χρησιμοποιηθούν και σε τι βαθμό εξαρτάται από την policy των δύο άκρων.

## Goal 4: Scalability

Is web scalable (in each of the three dimensions)? How is scalability at each dimension achieved, that is, which are the scalability techniques used along each dimension?

### Scalability

#### 1. Size.

- a. Το web είναι πλήρως scalable όσον αφορά το αριθμό των χρηστών και των διεργασιών που μπορούν να συνδεθούν σε αυτό.

(Σημείωση: αν δούμε την εξάρτηση του web από τα ip, θα δούμε ότι ο αριθμός των δυνατών ip's είναι πεπερασμένος. Άλλα αυτό δεν αποτελεί ουσιαστικό πρόβλημα scalability γιατί αφενός ο αριθμός των ip είναι πολύ μεγάλος και αφετέρου το πρόβλημα αυτό λύνεται με διάφορες τεχνικές (dynamic ip's, multiple dns names στην ίδια ip κτλ κτλ...))

- b. Η τεχνική που χρησιμοποιείται είναι το distribution. Οι χρήστες δεν συνδέονται όλοι στο ίδιο μηχάνημα άλλα σε πολλά μηχανήματα στο δίκτυο.
- c. Replication ώστε να έχουμε load balancing λόγω των πολλών χρηστών.

#### 2. geographical.

- a. Δεν υπάρχει περιορισμός. Μπορούν να συνδεθούν νέα ανεξάρτητα domains στο web. Δεν υπάρχει γεωγραφικός περιορισμός στα domain αυτά (world-wide-web...). Μπορεί θεωρητικά να συνδεθεί ένας server που βρίσκεται στην Αμερική με το όνομα www.domain\_name.gr...
- b. Η τεχνική που χρησιμοποιείται είναι το distribution με την χρήση του DNS.

#### 3. Number of administrative domains.

- a. Μπορούμε να έχουμε ΟΣΑ domains θέλουμε. Το μόνο που χρειάζεται είναι η αντιστοίχιση του domain name στον αντίστοιχο DNS server. Για την προστασία από κακόβουλους χρήστες στο πρωτόκολλο του DNS ορίζεται μία ιεραρχική δομή με τους authoritative dns servers.
- b. Η τεχνική που χρησιμοποιείται είναι το distribution με την χρήση του DNS.