Ciências ULisboa

Programação em Sistemas Distribuídos

MEI-MI-MSI

2018/19

4. Advanced Distributed Systems Services

Prof. António Casimiro

Distributed Systems Services

- Distributed File Services
  - (NFS,AFS,CODA,GFS)
- **Name and Directory Services**
  - **(X.500)**
- Time Services
  - (NTP)

Name and Directory Services

Name and directory services

- Name and directory services:
  - To **identify services and users independently from their localization**: dynamically establish a binding between name and localization (address)
  - Directory services more information than just names: they allow **imprecise** and/or **functional queries** (e.g., find all laser printers, find PCs running Linux)
- Internet DNS
  - Name service, which maps domain and sub-domain names in IP addresses; hierarchical service
  - **Example**: gcc.alunos.di.fc.ul.pt resolves to an address by iteratively querying the involved domain and sub-domain servers: .pt →.ul → .fc → .di → gcc

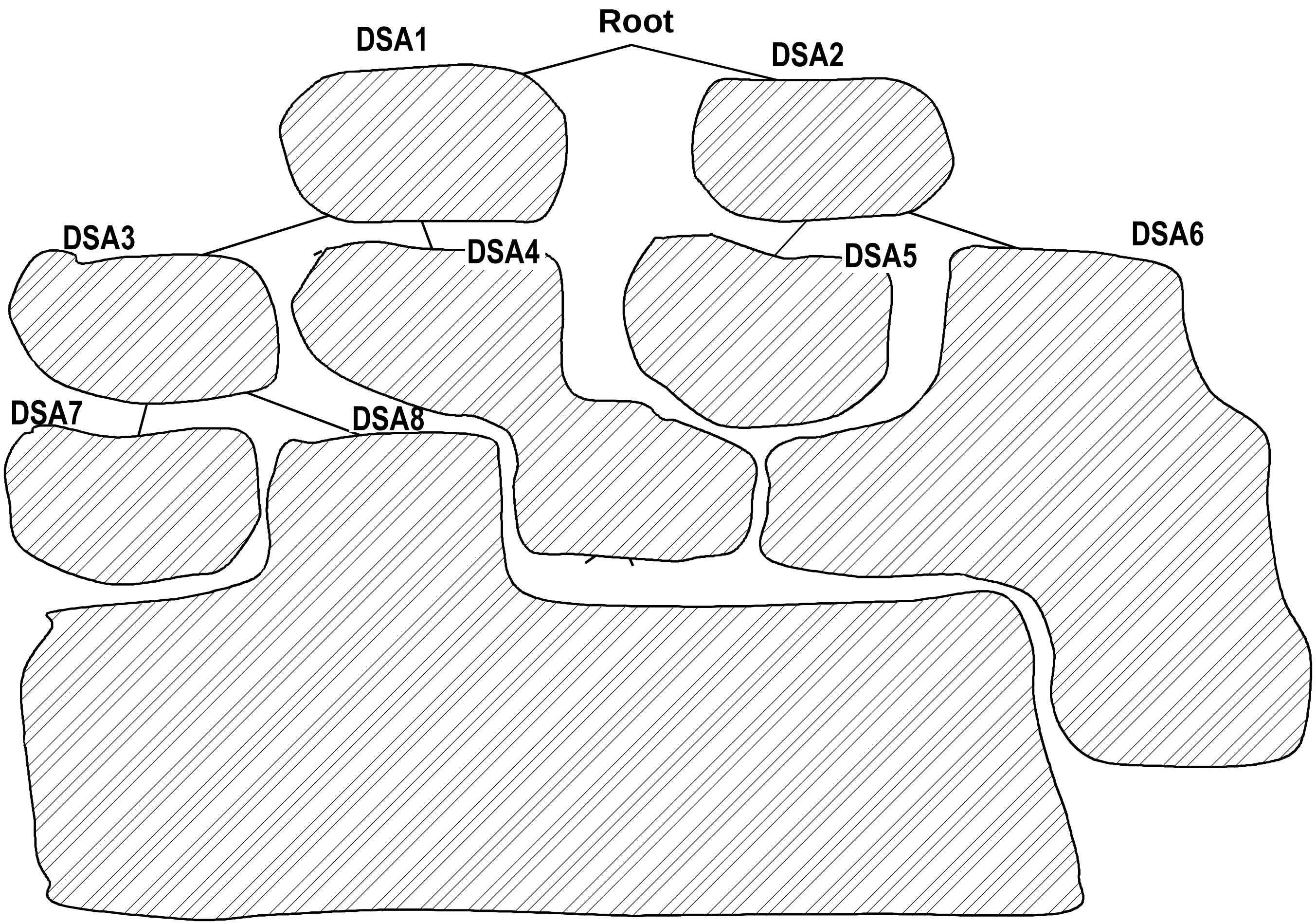
X.500 names

- A name serves to refer to services and users
- A name must be composable and unique
- X.500 name:
  - Distinguished Name (DN)
    - Ordered sequence of Relative Distinguished Names (RDN)
  - Relative Distinguished Name (RDN)
    - Non-ordered set of attributes, with well-defined types
    - Names with context (e.g., countries) chosen from normalized codes whenever possible (e.g., country codes with two digits ISO 3166)
- Name construction:
  - Attribute definition
    - Example: C (country); O (organis.); U (org. unit); CN (name)
  - RDNs are defined by giving values to each attribute
    - Example: C=PT; O=ULFacCien; U=DpInf; CN=Beto
  - Then DNs can be constructed
    - Example: DN Beto na FCUL → C=PT/O=ULFacCien/U=DpInf/CN=Beto

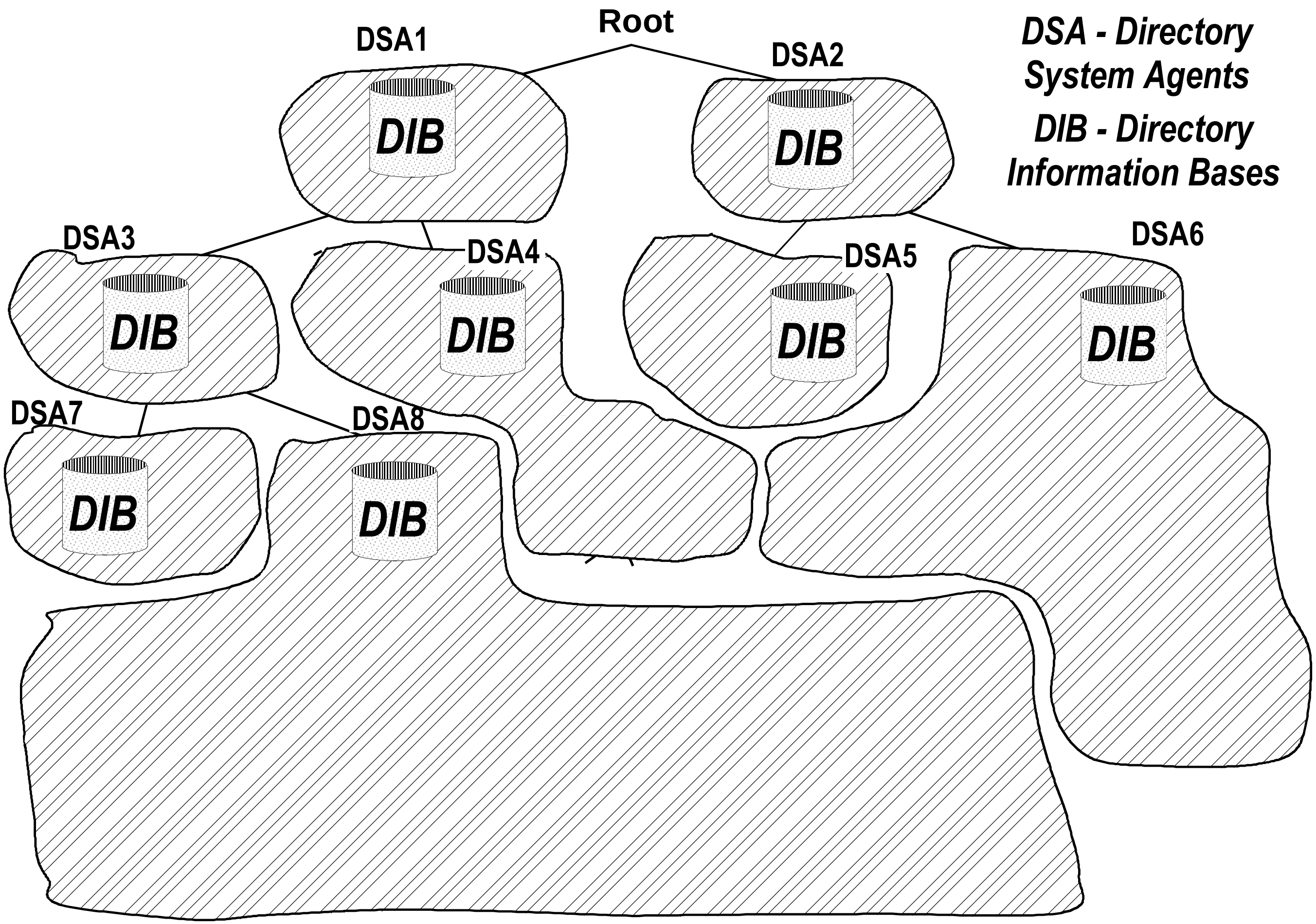
Name organization in X.500

- DNs can be organized in a global tree:
  - RDNs are assigned by hierarchically organized agents
  - Each agent ensures that assigned names are not ambiguous and are unique in its area
  - Then, the name is composed and thus scalable, and unique in the entire system (because it is composed by locally unique names)
- DNs usually represented in a friendly way, hiding attributes
  - Example: C=PT/O=ULFacCien/U=DpInf/CN=Beto → PT.ULFacCien.DpInf.Beto
- This organization builds on the X.500 directory service
  - The tree is the Directory Information Tree (DIT)
  - The agents in each node are Directory Service Agents (DSA)
  - DSAs act over a Directory Information Base (DIB)

X.500 directory service
Directory information tree



X.500 directory service
Directory information tree

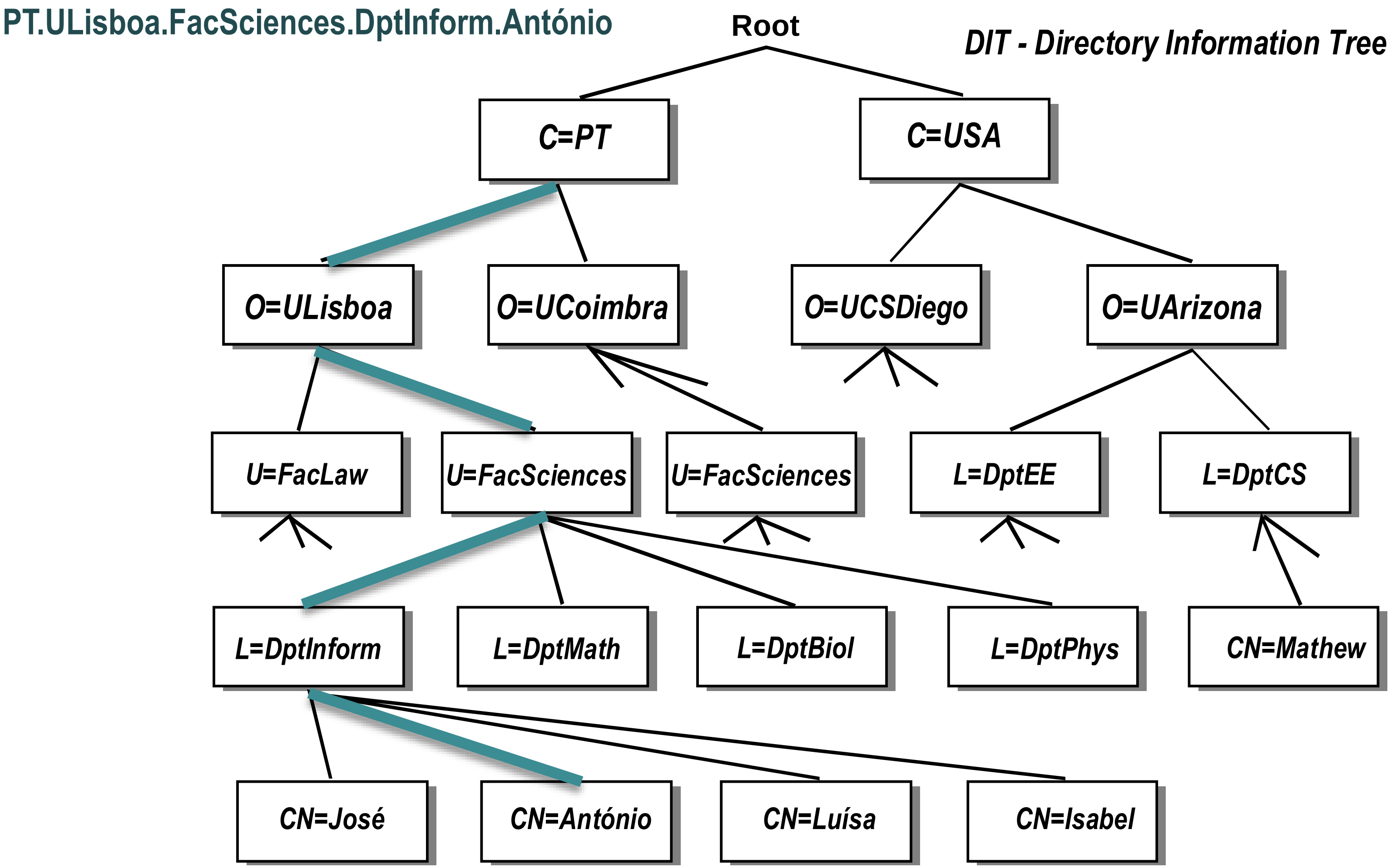


DSA - Directory System Agents

DIB - Directory Information Bases

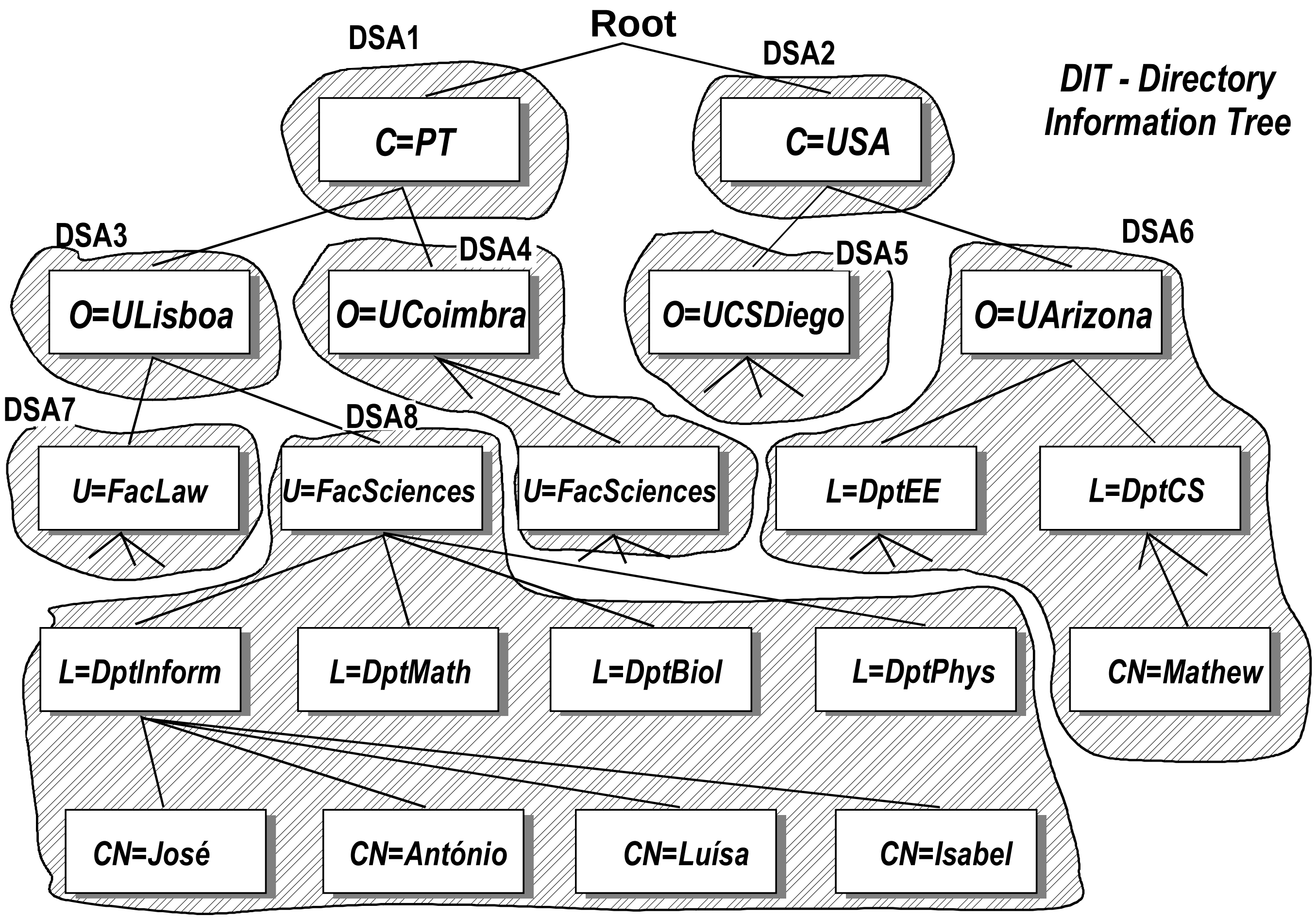
X.500 directory service

DNs tree



X.500 directory service

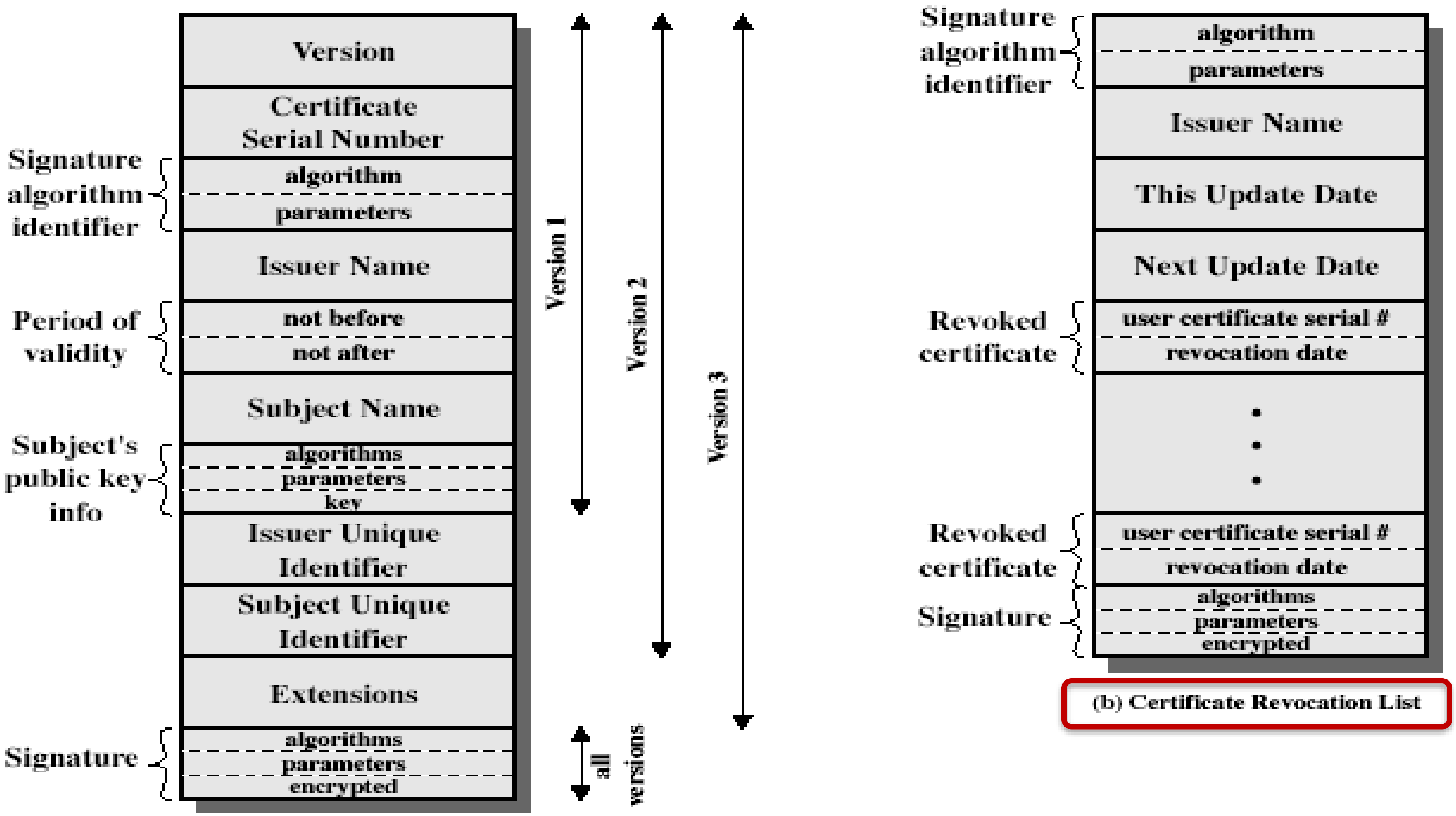
Mapping of DNs on DSAs



X.509 authentication service

- Part of the X.500 CCITT standard
- Defines a framework for authentication services
- A Certificate Authority (CA) issues certificates binding a public key to a DN (X.500)
- A directory may contain public key certificates
  - Containing user's public keys
  - Signed by a certifying entity
- Also defines authentication protocols
- Uses public key cryptography and digital signatures
  - Normalized algorithms, but RSA is recommended
  - May be used in many contexts: email security, IP security, Web security

Structure of a X.509 certificate



(a) X.509 Certificate

(b) Certificate Revocation List

X.509 certificate
Example

```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
                OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                   (…)
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
       (…)
```

Distributed Systems Services

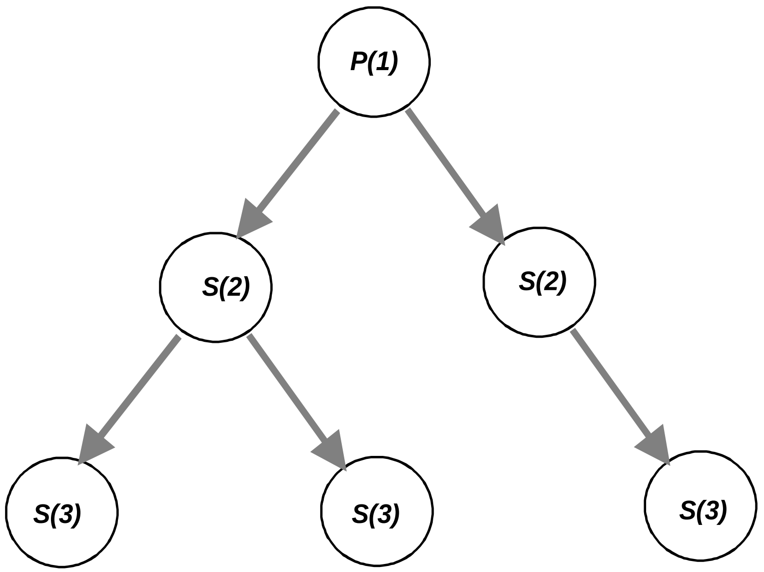
- Distributed File Services
  - (NFS,AFS,CODA,GFS)
- Name and Directory Services
  - (X.500)
- **Time Services**
  - **(NTP)**

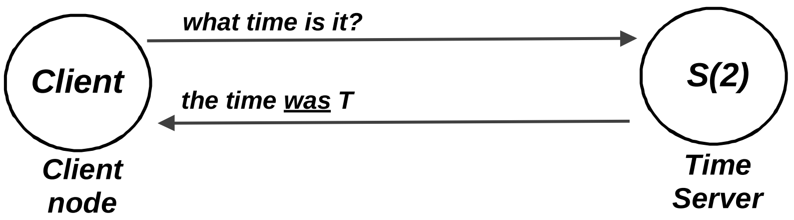
Global Time Services

Global Time Services

Case study: Network Time Protocol

- Network Time Protocol (NTP):
  - **Standard Internet time service**
- Some characteristics:
  - Provides UTC time
  - Resilient to connectivity problems
  - Some protection against attacks (e.g., spoofing) by authentication
  - **Average accuracies in the order of the tens of milliseconds**





Global Time Services
Case study: Network Time Protocol

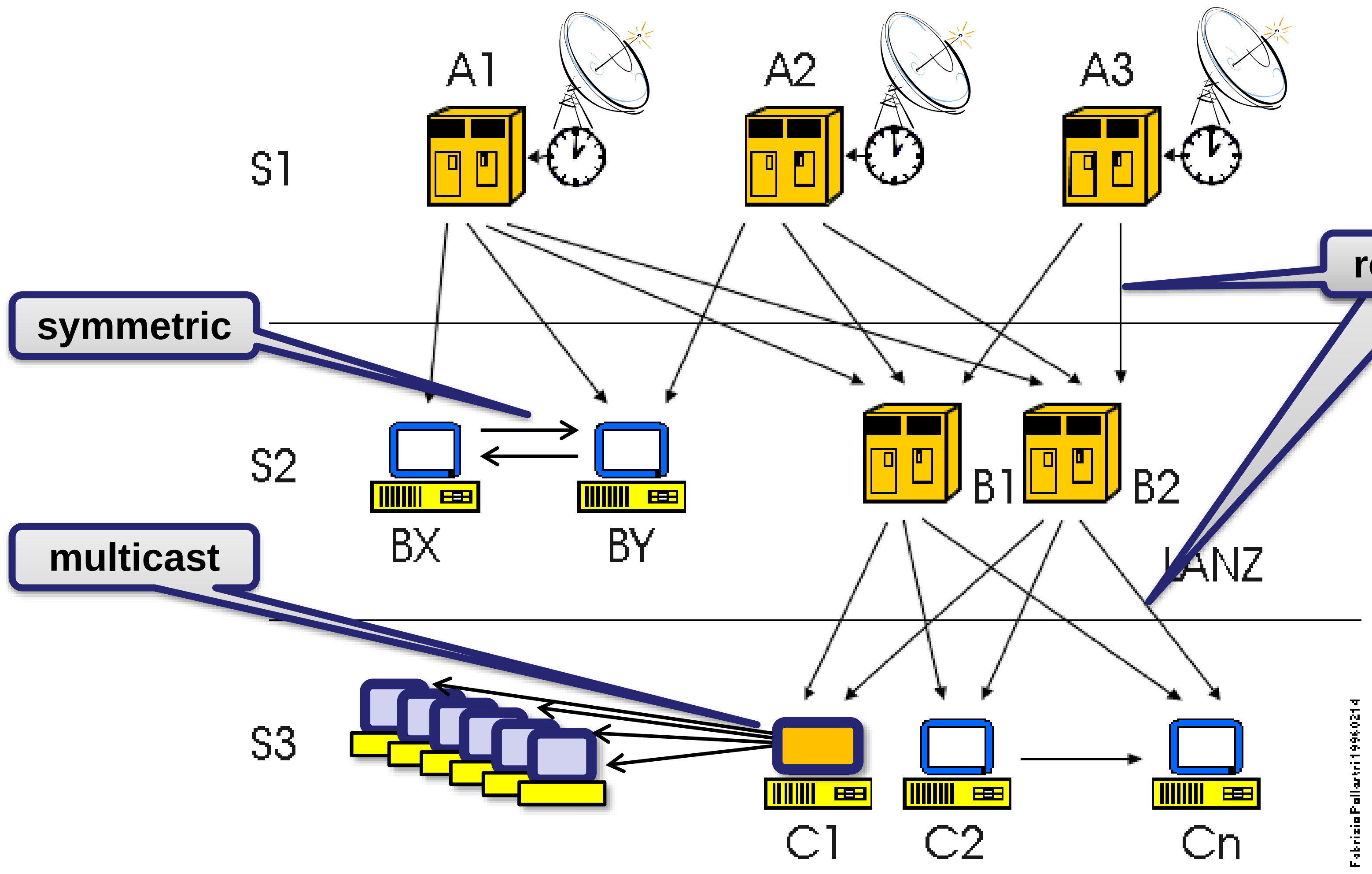
- NTP Time Synchronization Service:
  - Hybrid **hierarchical tree structure**
  - Different layers (strata) use different synchronization schemes
  - Clock servers organized in descending order of intrinsic accuracy in the hierarchy (degrades from top to bottom)
- Strata hierarchy:
  - **Stratum 1 (top) – primary servers**: directly synchronized to external UTC-compliant time references (e.g., GPS, atomic time sources)
  - **Stratum 2 to n – e.g. stratum 2, secondary servers**: directly synchronized to n-1 stratum server time references

Global Time Services

Case study: Network Time Protocol

- Several synchronization modes
  - **Multicast**:
    - Simplest and least effective, works acceptably inside high-speed (low-delay) LAN networks with multicast (e.g. datacentres): one more or more servers, sync'd from stratum above, simply multicast their time to servers inside infrastructure LANs
  - **Round-trip**:
    - Most generic, used at the edges (lower strata): inspired by Cristian's master-slave round-trip clock synchronization protocol, probabilistically achieves better synchronization
  - **Symmetric**:
    - Used at upper strata, whenever it is desired to improve the accuracy: by symmetric message exchanges, whereby servers of the same stratum or adjoining strata improve their synchronization through agreement-based adjustments

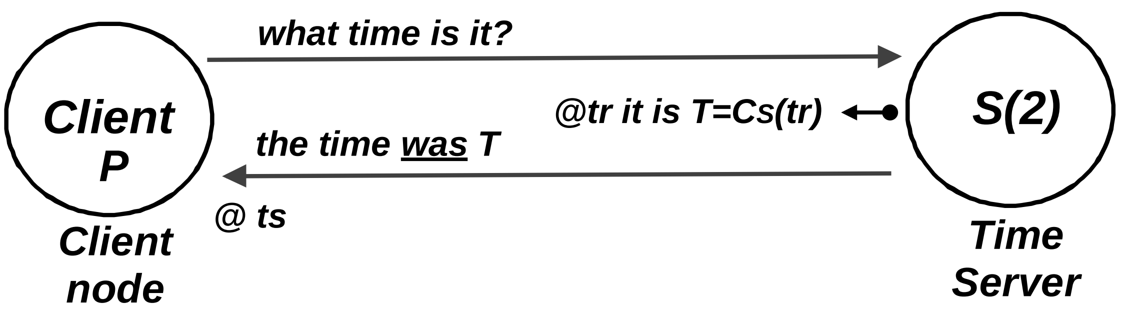
NTP



A1
A2
A3
S1
round-trip
symmetric
S2
B1
B2
BX
BY
multicast
LANZ
S3
C1
C2
Cn

NTP clock synchronization
Round-trip synchronization

- External synchronization:
  - Round-trip: based on reading from a master clock
- **Problem**:
  - When the response (T) arrives from the server, the time at S "was T"
  - How to adjust P's clock @ts, with the best estimate of the time it is at S, @ts ?!

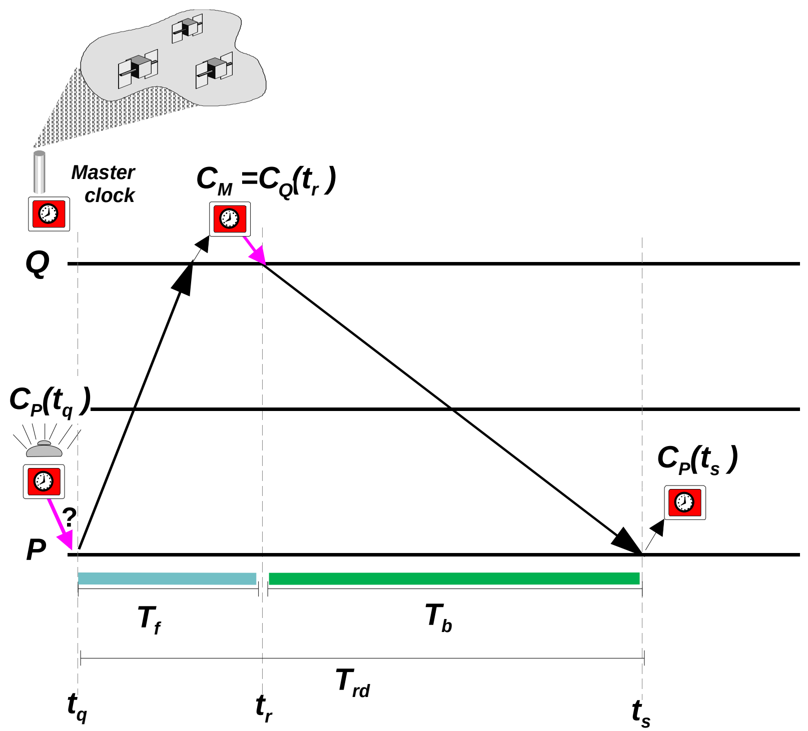


NTP clock synchronization
Round-trip synchronization

- Adjusting clocks:
  - T_f and T_b not known
  - Measure round-trip T_{rd} at P: $T_{rd}=C_p(t_s)-C_p(t_q)$
  - Estimate t_r to be at midpoint, so estimate $T_b, T_f \approx T_{rd}/2$
  - Adjust local clock C_P to received timestamp plus T_b:

$$C_p(t_s)=C_Q(t_r)+T_b=C_Q(t_r)+T_{rd}/2$$

- But round-trips are seldom symmetric, which causes an error when estimating $C_p(t_s)$

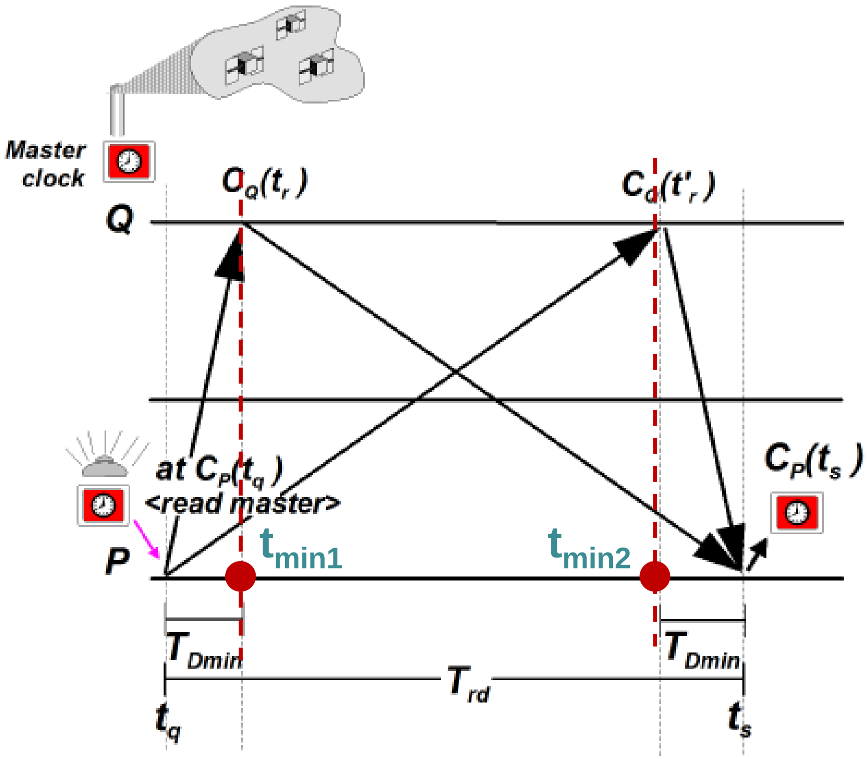


- **Can we bound the error?**
- **Is it possible to minimize it?**

NTP clock synchronization
Estimating accuracy error

- Estimating accuracy error:
  - Error depends on the symmetry of the round-trip transmission
  - Symmetric round-trip: excellent accuracy
  - But we don't know when, except when $T_{rd}=2T_{Dmin}$ (minimum tx delay, known)
  - We use an indirect technique
- Foundation:
  - Boundaries of arrival of P time request at Q, and departure of Q reply:
  - t_{min1} : $t_q + T_{Dmin}$
  - t_{min2} : $t_s - T_{Dmin}$
  - When the timing msg arrives (@t_s), $C_Q(t)$ contained therein marks an instant between:
  - $T_{early} = t_q + T_{Dmin}$
  - $T_{late} = t_q + T_{rd} - T_{Dmin}$
  - **Accuracy error** when C_P adjusted to the midpoint of the round-trip interval:

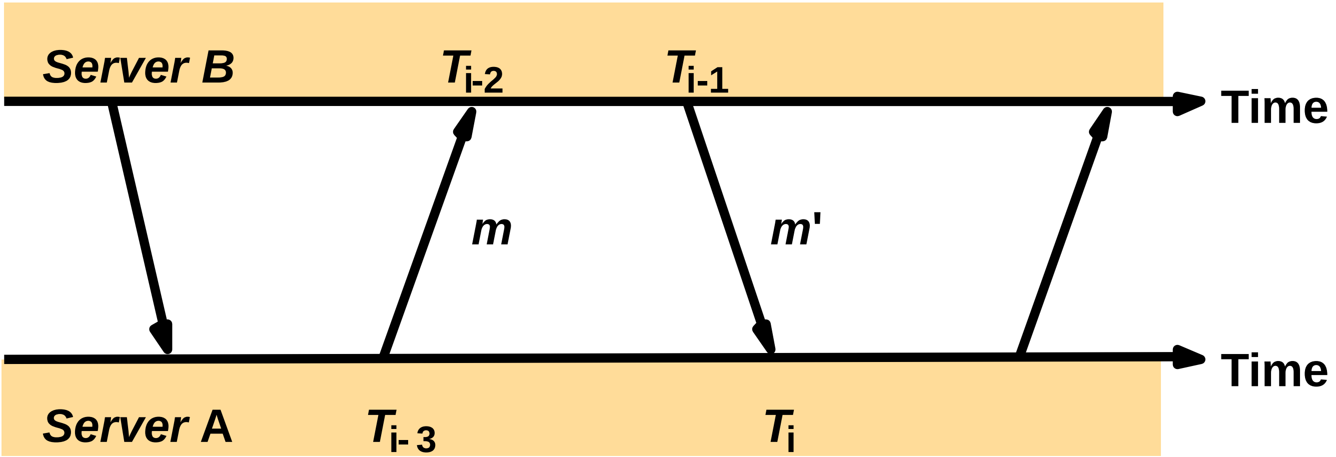
$$\varepsilon \leq \pm (T_{late} - T_{early})/2 = \pm (T_{rd}/2 - T_{Dmin})$$



NTP clock synchronization

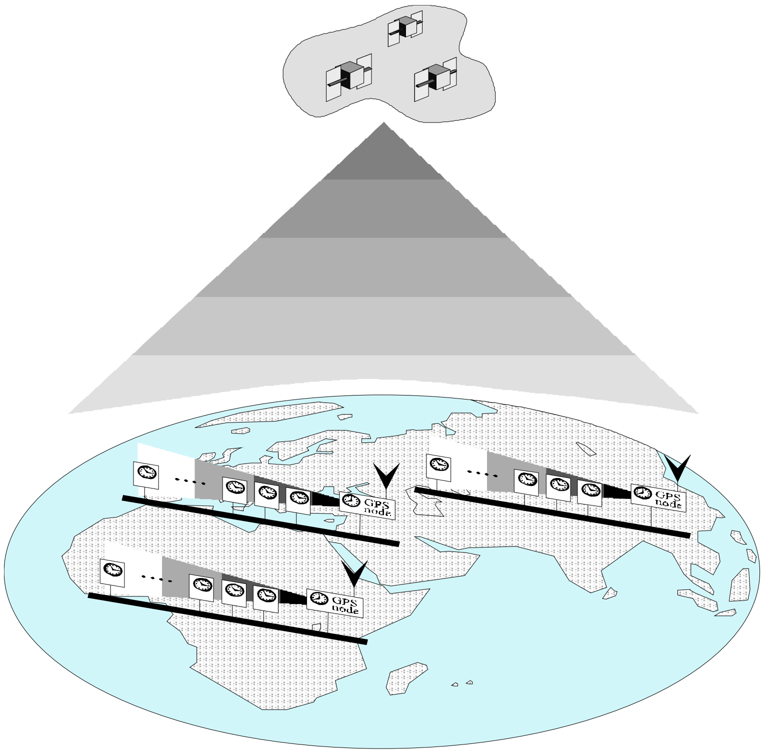
Minimizing accuracy error

- At the end of each roundtrip pair of msgs exchanged, ***A*** knows its own timestamps (T_{i-3}, T_i) and ***B***'s timestamps (T_{i-2}, T_{i-1})
- ***A*** computes <o,d> : an estimate of the offset of ***A*** and ***B*** clocks, and of the total transmission delay
- Several tries are performed, and from the last n <o,d> pairs, the offset o_i of the pair having the minimum delay d_i is chosen for the clock adjustment
- Also, ***A*** talks to more than one server ***B*** and performs peer-selection based on: lower filter dispersion; lower stratum n_r



Global time services
Case study: CesiumSpray

- Large-scale
- Based on GPS
- Highly precise
- Highly accurate
- Scalability due to hierarchical structure:
  - Wireless on global part
  - LAN on local part



CesiumSpray

Hybrid *a posteriori* clock synchronization

- Hybrid:
  - Internal/external
  - *a posteriori*/GPS
- Precision:
  - Candidate virtual clocks started simultaneously
  - Agreement on clock made a posteriori (residual interference on precision)
- Accuracy:
  - Vector with clock readings
  - Selection of best clock:
    - Mean value (internal sync)
    - GPS-clock (external sync)