telespazio
una Società Finmeccanica

CAC

Connection Admission Control

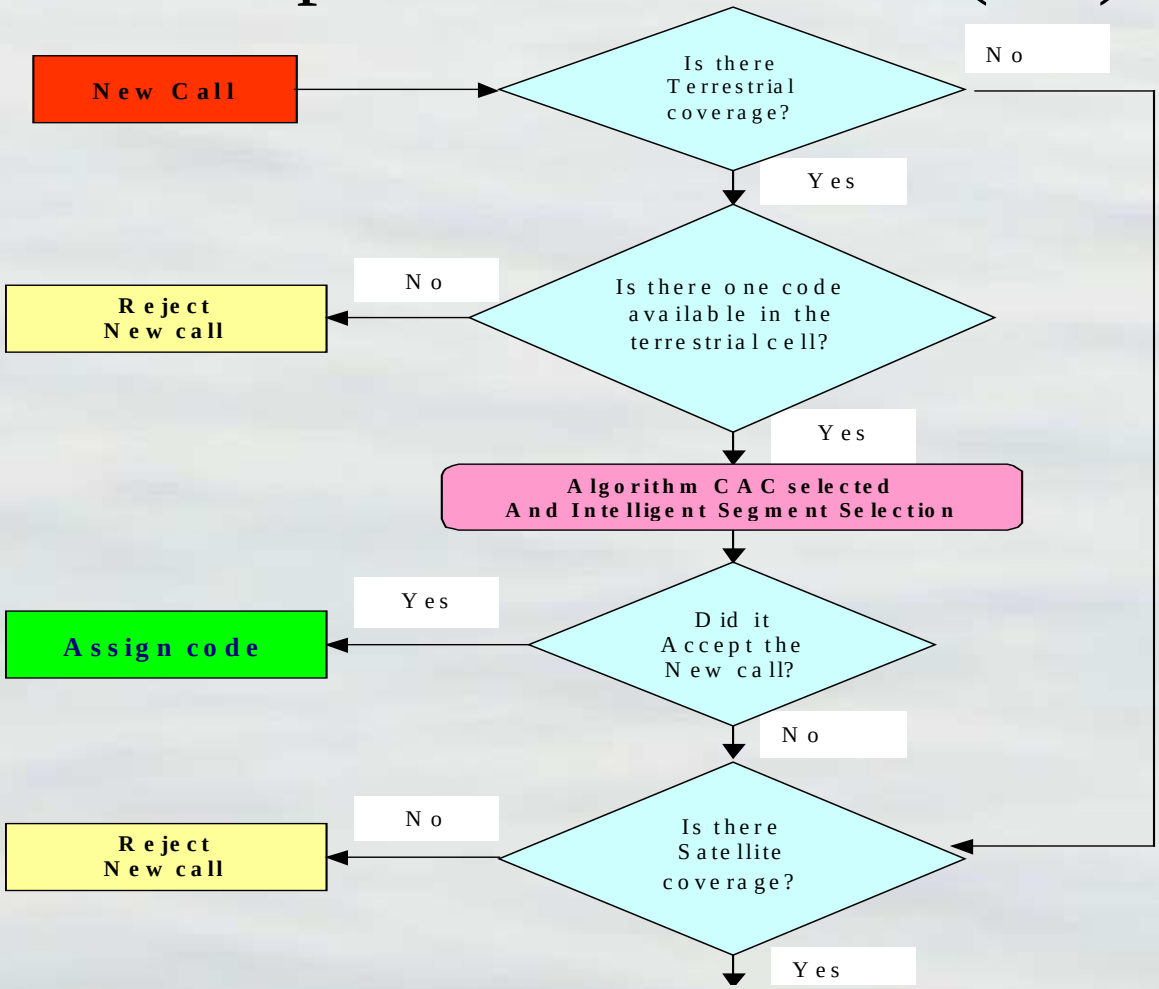
Bergeggi, October 28, 2004

Speaker: Rosario Toscano (rosario_toscano@telespazio.it)

CAC Features

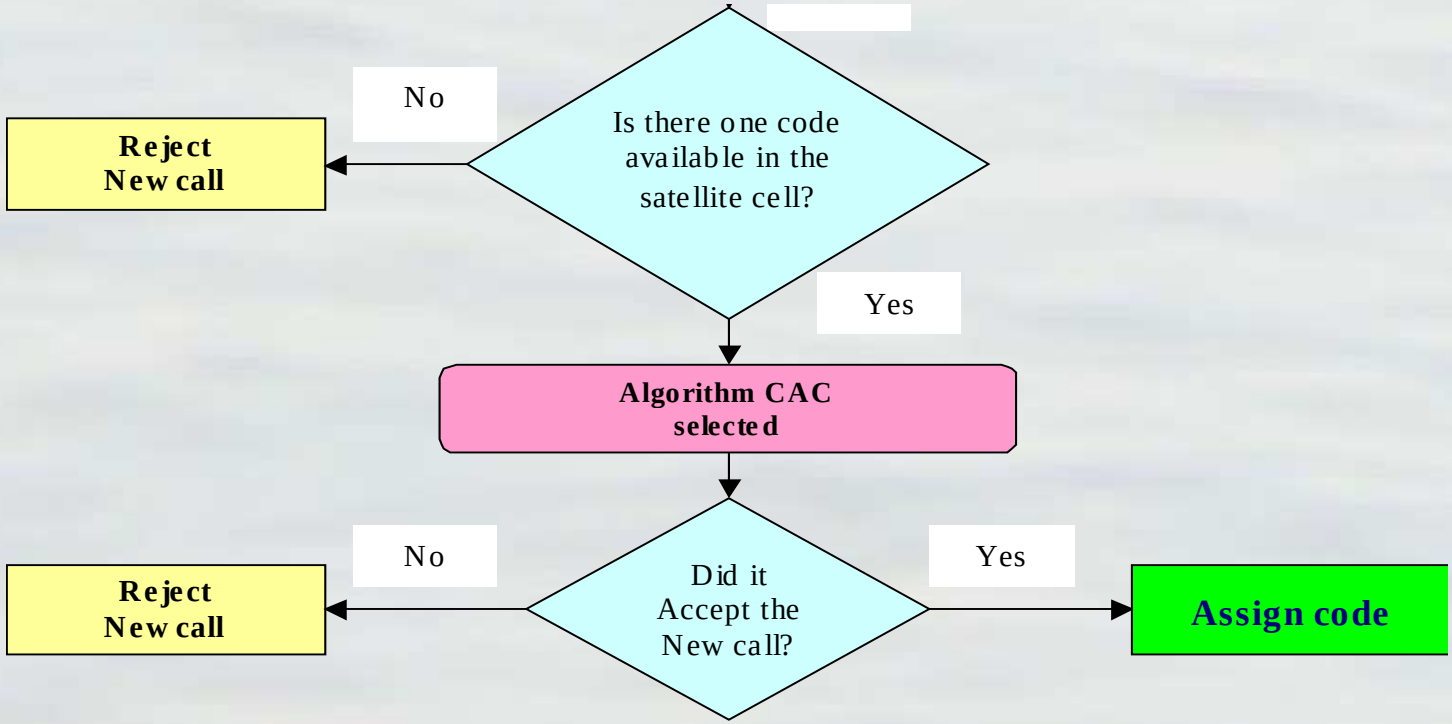
- Implementation of advanced resource optimisation (Up-link, Down-link) algorithms in an admission control procedure
  - QoS management of different services (conversational, streaming, interactive and background)
  - Resource reallocation management
  - Intelligent Segment Selection (ISS)
  - Code assignation policy
- Implemented in the DANS framework

CAC Operative Flowchart (1/2)



Continue ...

CAC Operative Flowchart (2/2)



Algorithms Implemented for the CAC

- Up-Link CAC Algorithms

- Ø ArrowsCacBase
- Ø ArrowsCacReArrangement
- Ø ArrowsCacReArrTDU

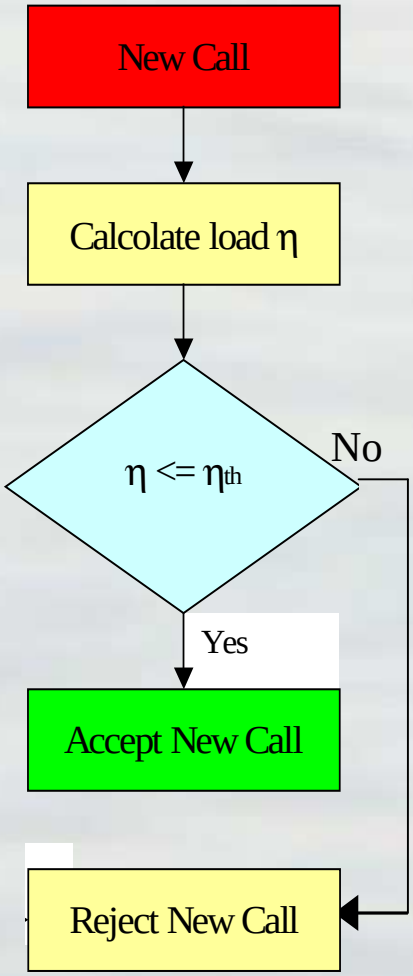
- Down-Link CAC Algorithm

- Ø PowerBasedCAC

- Code Assignation Strategy

- Ø RandomStrategy
- Ø LeftmostStrategy
- Ø CrowdedStrategy

ArrowsCacBase (UpLink)



Statistical calculation of the load cell

$$\eta = (1+f)\sum_{i=1}^{N} \frac{1}{\frac{SF_i}{\left(\frac{E_b}{N_0}\right)\nu_i}+1} + (1+f)\frac{1}{\frac{SF_{N+1}}{\left(\frac{E_b}{N_0}\right)\nu_i}+1} \leq \eta_{th}$$

SF **spreading-factor**

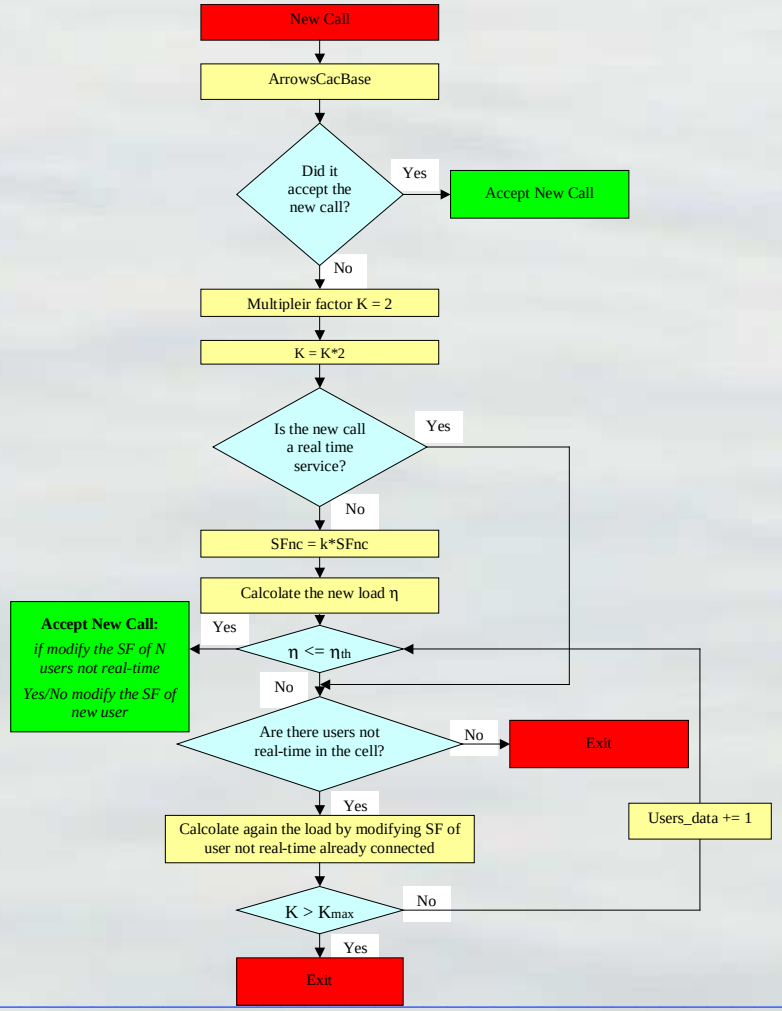
Eb/N0 **quality of service**

f **fraction of the own-cell received power due to other-cell interference power**

ν **activity factor of the source**

η th **load threshold**

ArrowsCacReArrangement (Up-Link)



This algorithm calls:

Ø ArrowsCacBase function

If the ArowsCacBase function decides that the new call can not accede to the system, then the ArrowsCacReArrangment function starts to modify the bit-rate of some users (not real time service) already connected (by modifying the spread factor) in order to lower the cell load.

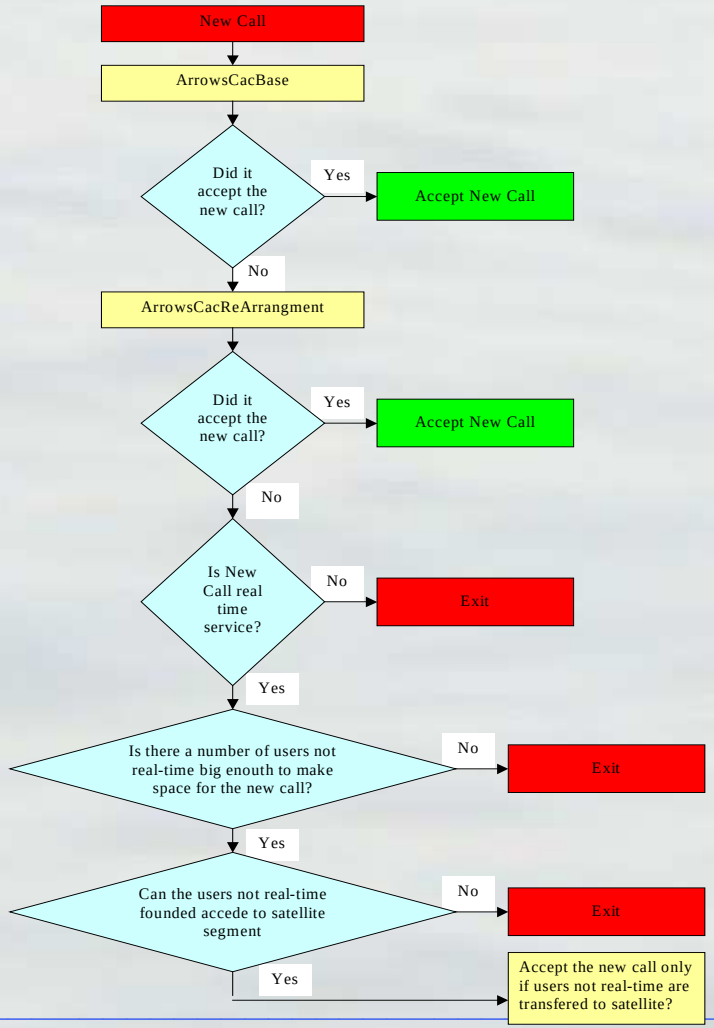
The algorithm suggests:

the number of users for which it is needed to modify the spreading factor

the multiplying factor necessary to modify this value (SF)

Moreover the algorithm suggests if it is needed_ or if it is possible to modify the spreading factor of the new user.

ArrowsCacReArrTDU (UpLink)



This algorithm calls:

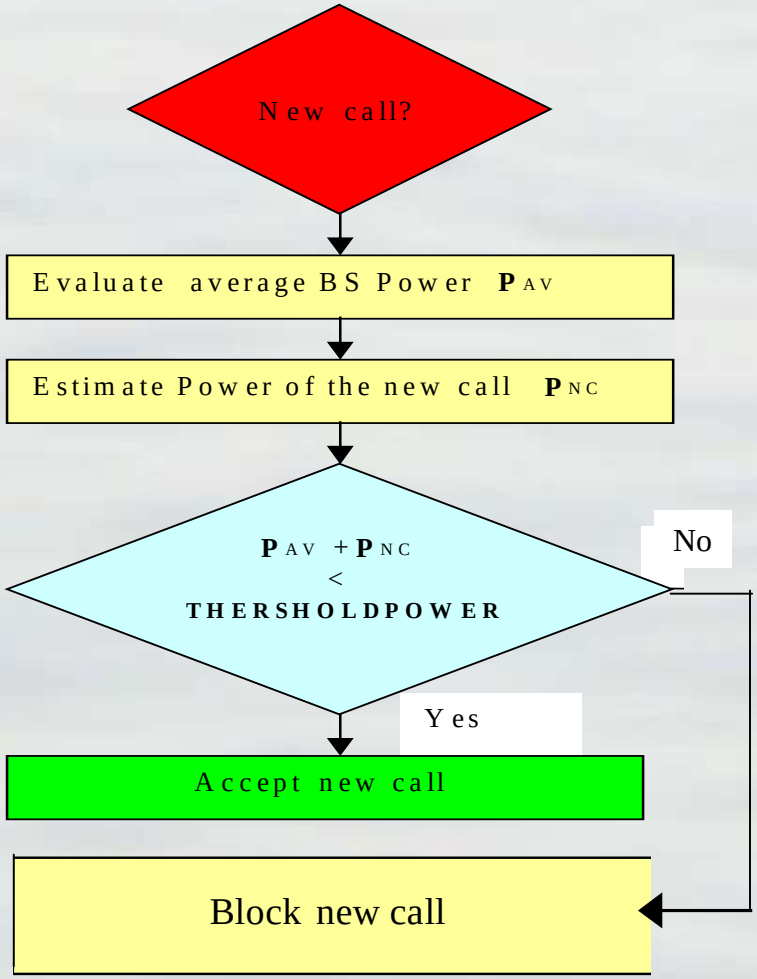
Ø ArrowsCACReArrangment

If the ArowsCacReArrangment does not accept the new request in terrestrial cell, it estimates if there is a satellite segment available. If there is a satellite segment and the type of service requested does not have particular requirements (i.e. not real time service) then the algorithm estimates the satellite load and decides whether to accet or reject the new call on this segment.

The algorithm suggests:

the number of users that is needed to transfer to satellite segment

PowerBasedCAC (DownLink)



The new call can accede to the system if:

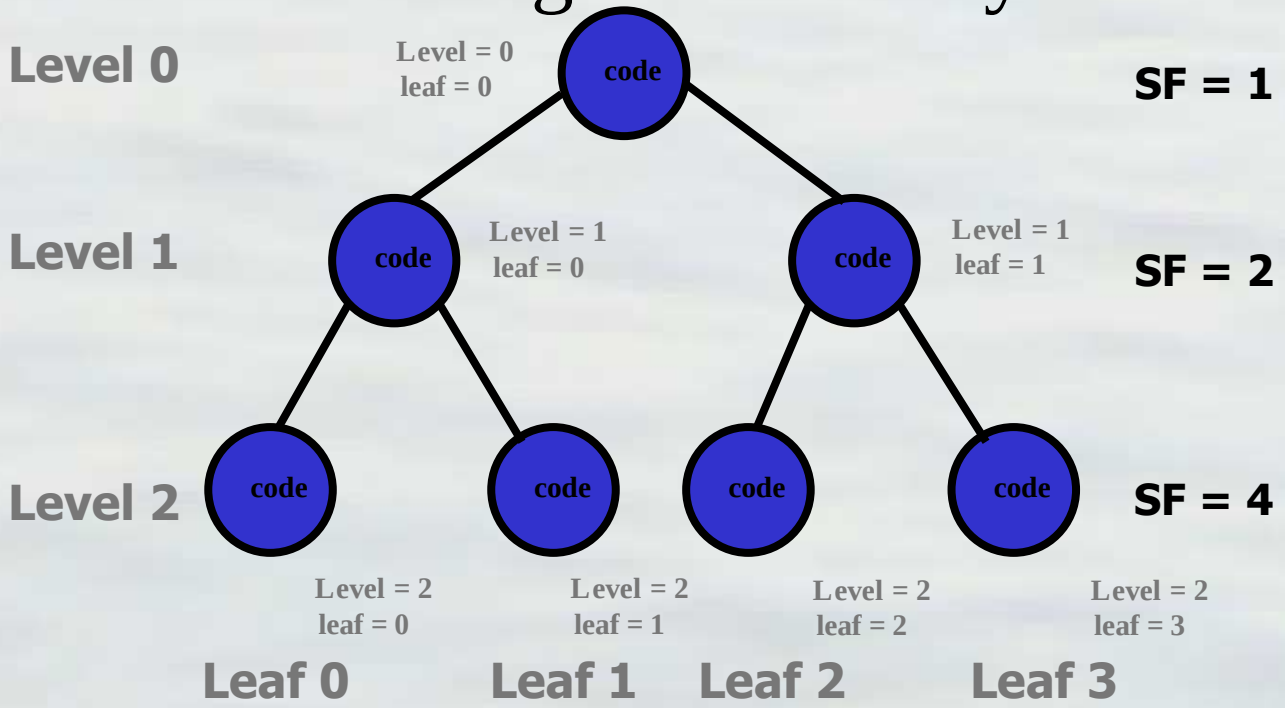
$$P_{AV} + \Delta P < P_{TH}$$

P_{TH} power threshold

P_{AV} for each cell the average power transmitted in a time windows with constant number of users

ΔP power assigned to a new user

Code Assignment Policy

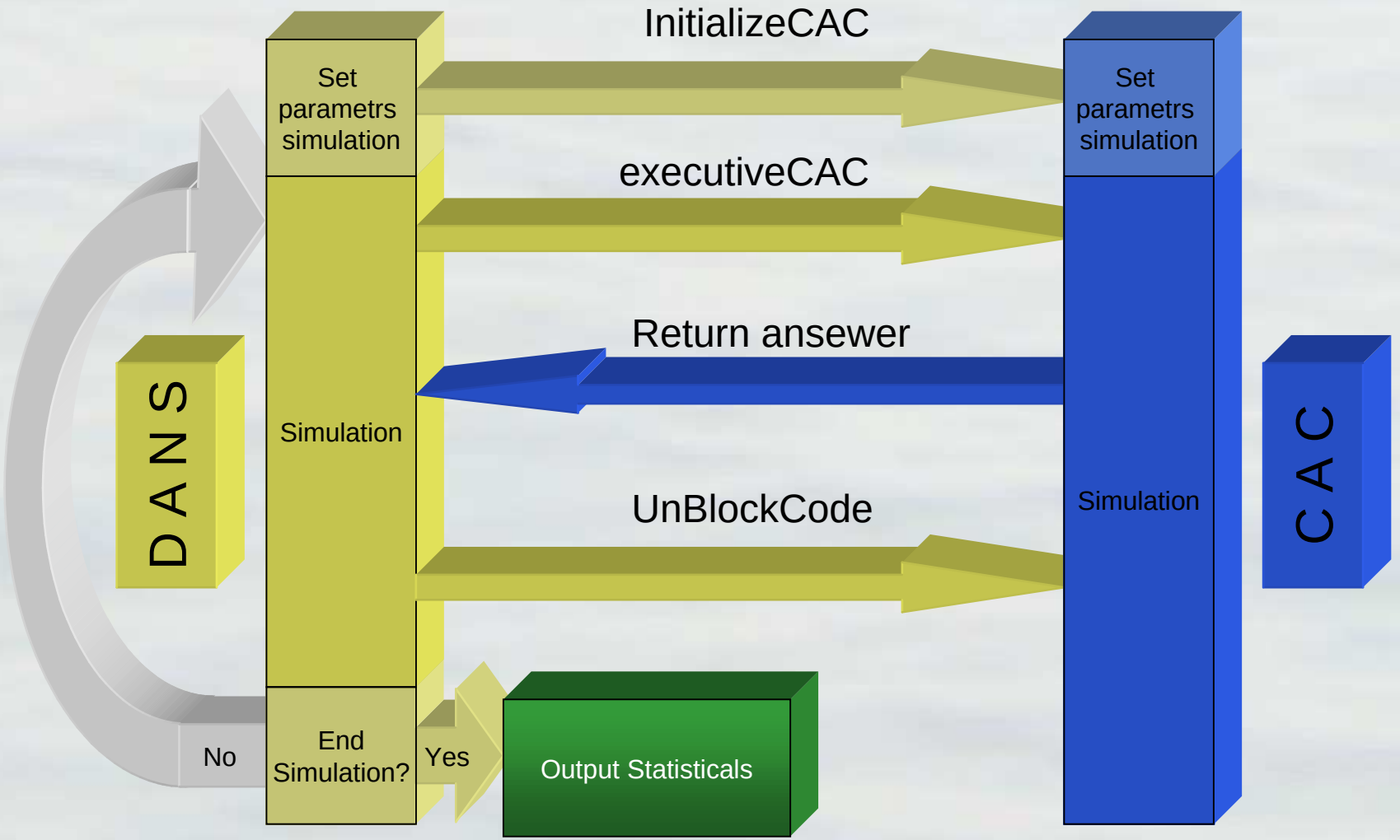


RandomStrategy – This function assigns the code to the new user by choosing randomly among the free leafs of the livel.

LeftmostStrategy – This function assigns the code to the new user by choosing the first free leaf of the level from left.

CrowdedStrategy – This function assigns the code to the new user by choosing among the free leafs of the level the one that is in the crowded tree's area. For every free leaf we calculate a crowding index. We choose the leaf with the bigger crowding index. If there are two or more leafs with the same crowding index then we choose the first one on the left.

DANS – CAC Interface (1/2)



DANS – CAC Interface (2/2)

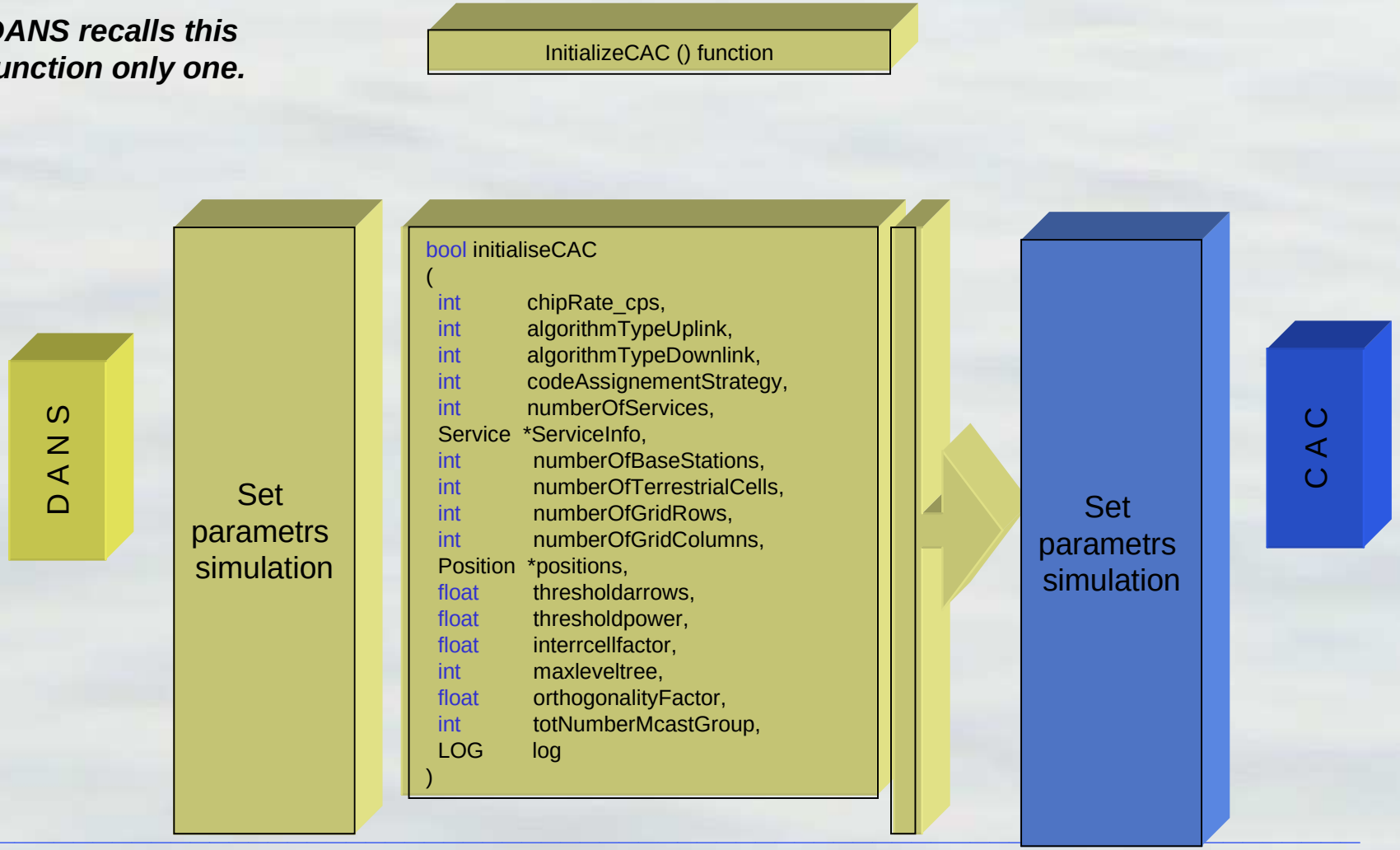
When a new call arrives the DANS calls a `cac( )` function

The CAC return to the DANS one of the following value:

Return value	CAC Policy
0	Not accept the new user
1	Accept the new user on Satellite/Terrestrial segment
2	Accept the new user under the load‘s rearrangement condition (by modifying the bit-rate of the users not real-time already connected)
3	Accept the new user on Terrestrial segment under the condition: transfer some users not real-time already connected from terrestrial segment to satellite segment

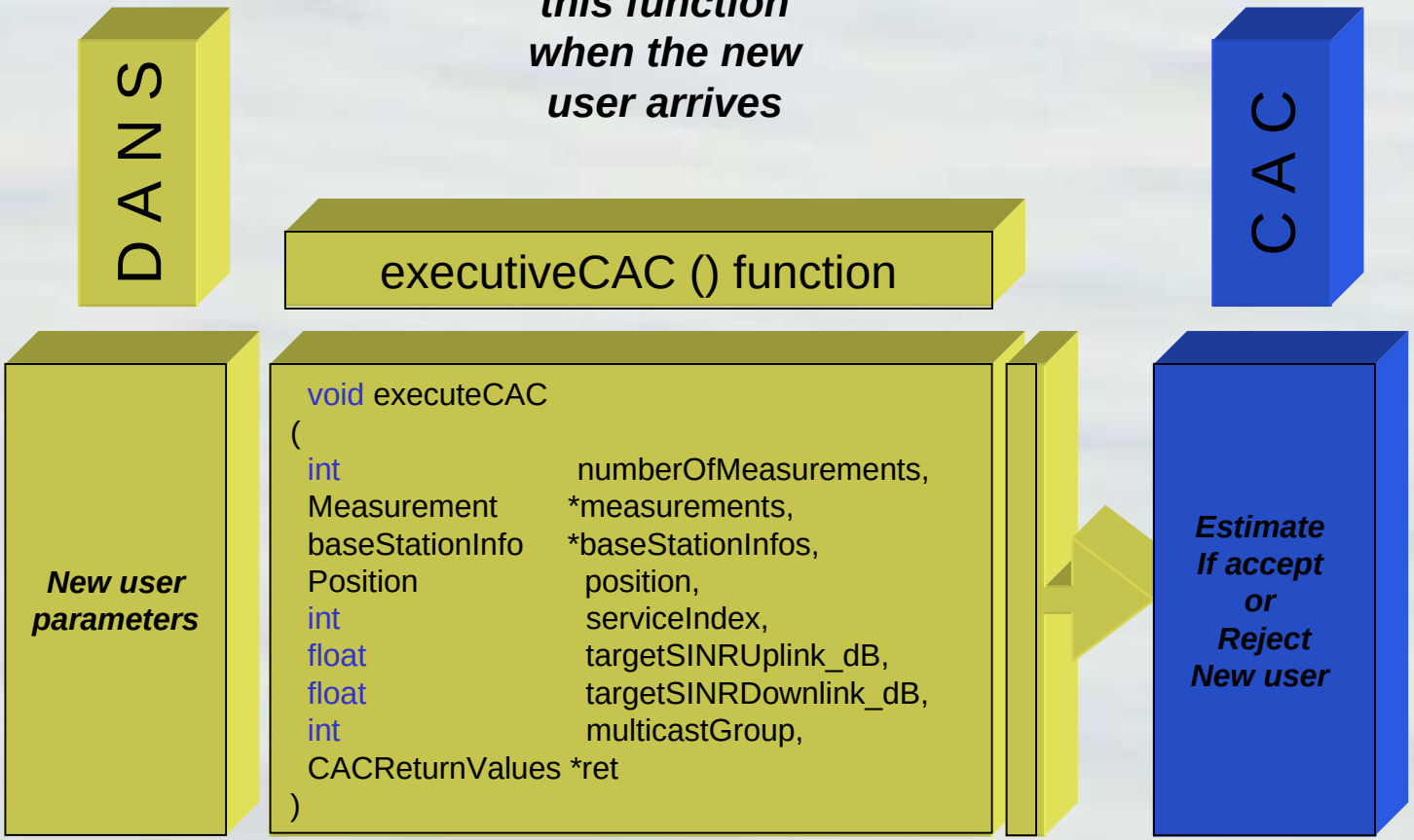
CAC () Functions (1/3)

DANS recalls this function only one.

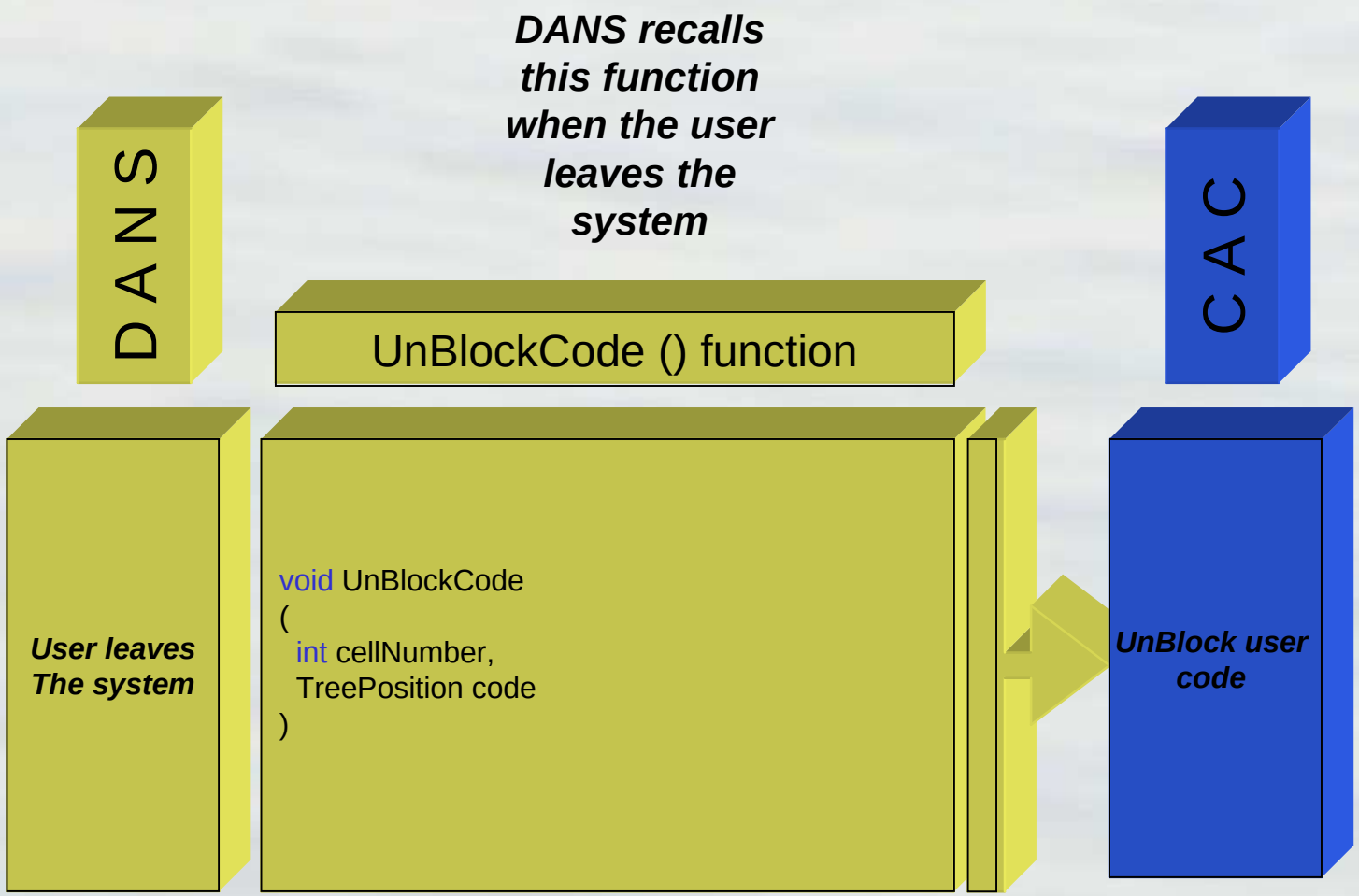


CAC () Functions (2/3)

DANS recalls this function when the new user arrives

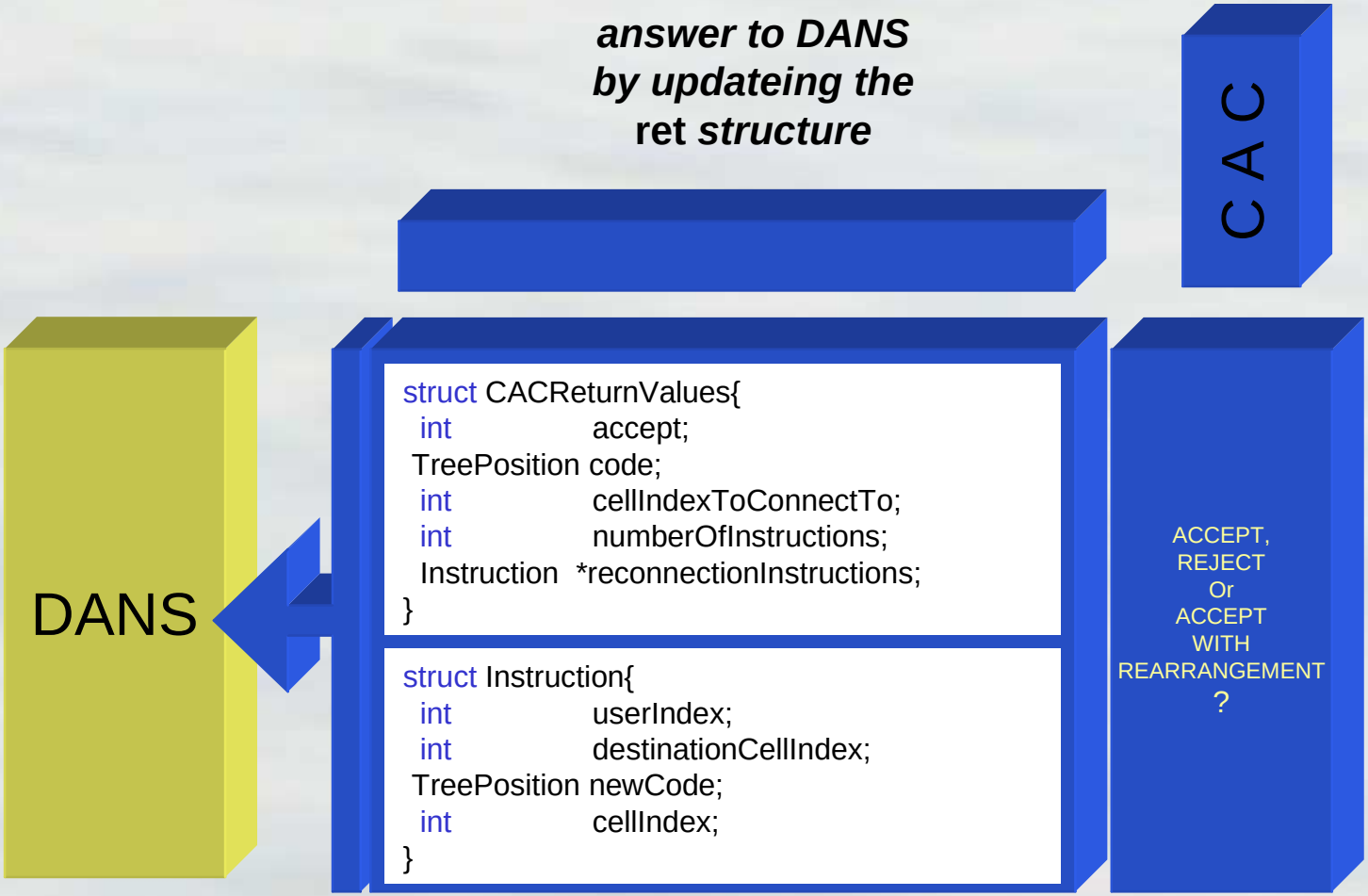


CAC () Functions (3/3)



Return answer

Cac returns his answer to DANS by updateing the* ret *structure



telespazio
una Società Finmeccanica

Thank You !!!