

Communication and Concurrency

Cinzia Di Giusto
cinzia.di-giusto@unice.fr

Concurrent systems

- multiple users that share services
- multiple users that cooperate
- PROTOCOLS





```
send "hello" to Charlie;  
receive ok from Charlie;  
send ok to Bob
```



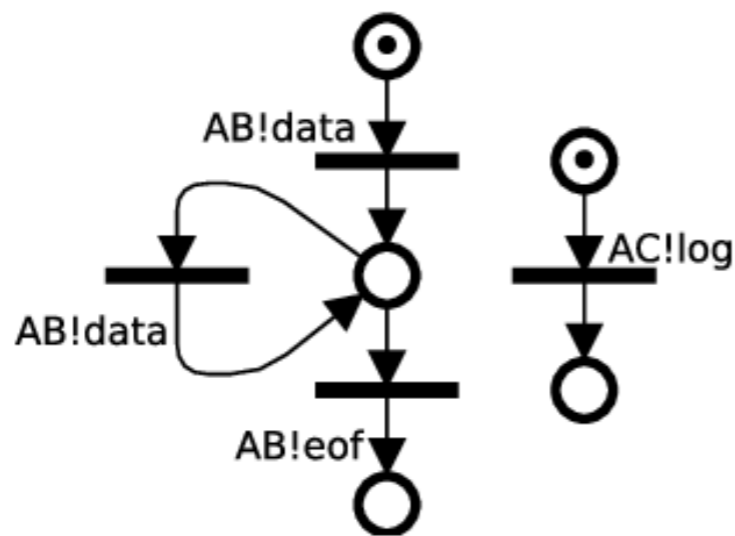
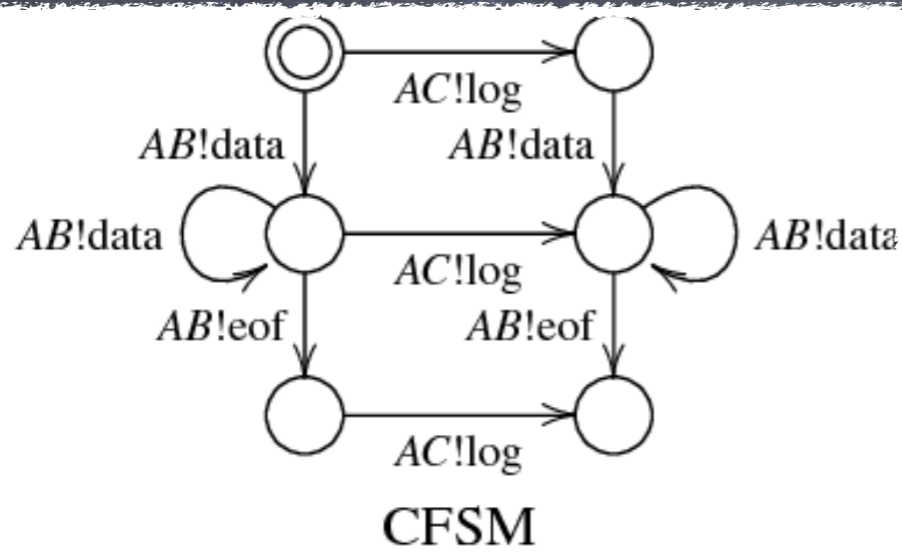
```
receive ok from Alice;
```



```
receive x from Alice  
if x then {  
  send ok to Bob;  
  send ok to Alice }  
else {  
  send ok to Alice;  
  send ok to Bob }
```

!Communication!

Abstractions



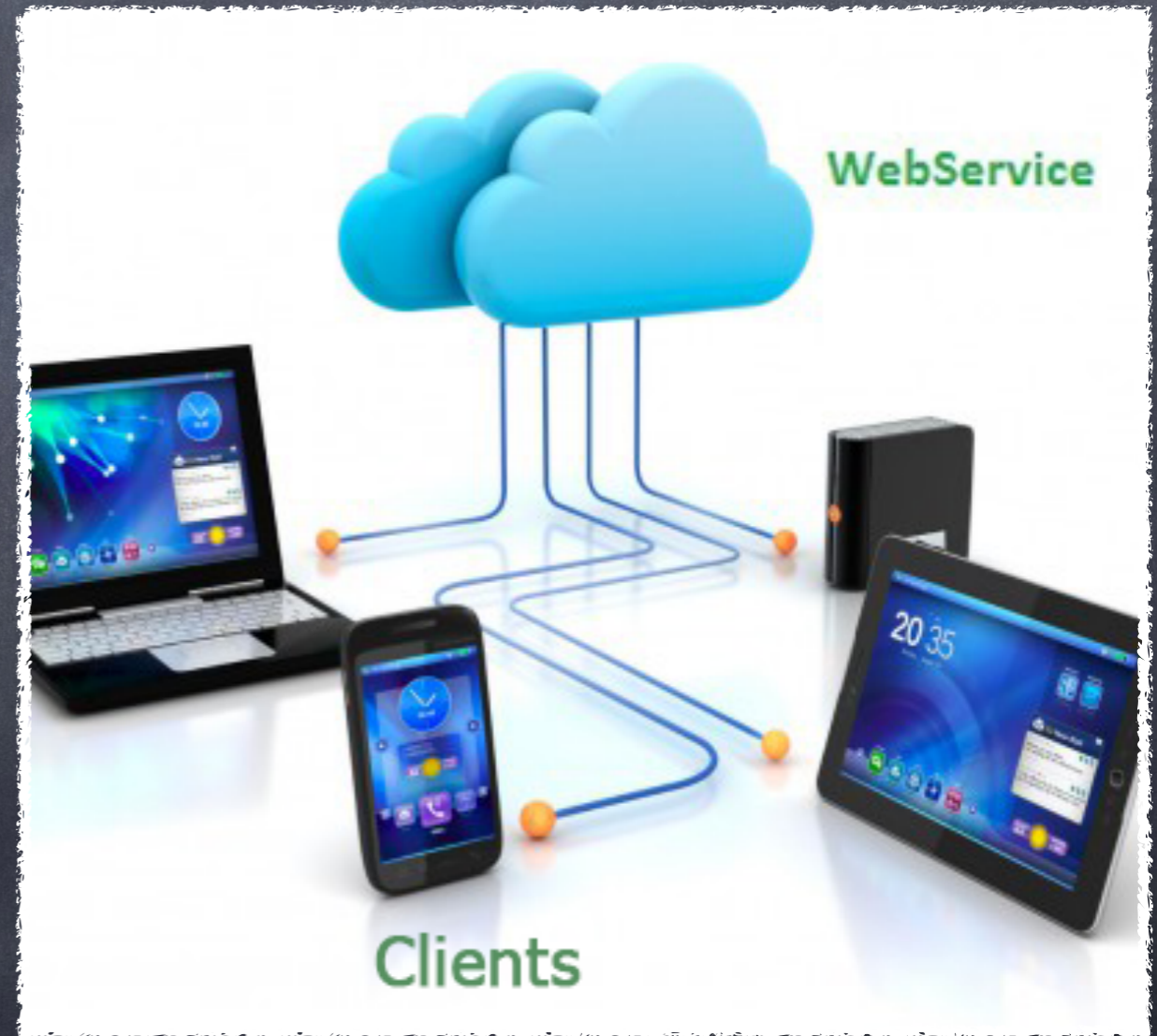
r A

- Petri nets
- Communicating automata
- Process algebra

Goal: good properties

Tools for:

- analysis
- verification





```
send "hello" to Charlie;  
receive ok from Charlie;  
send ok to Bob
```



```
receive ok from Alice;
```



```
receive x from Alice  
if x then {  
    send ok to Bob;  
    send ok to Alice }  
else {  
    send ok to Alice;  
    send ok to Bob }
```

See any problem?

ALICE SENDS A MESSAGE TO BOB
SAYING TO MEET HER SOMEWHERE.

UH HUH.

BUT EVE SEES IT, TOO,
AND GOES TO THE PLACE.

WITH YOU SO FAR.

BOB IS DELAYED, AND
ALICE AND EVE MEET.

YEAH?



I'VE DISCOVERED A WAY TO GET COMPUTER
SCIENTISTS TO LISTEN TO ANY BORING STORY.