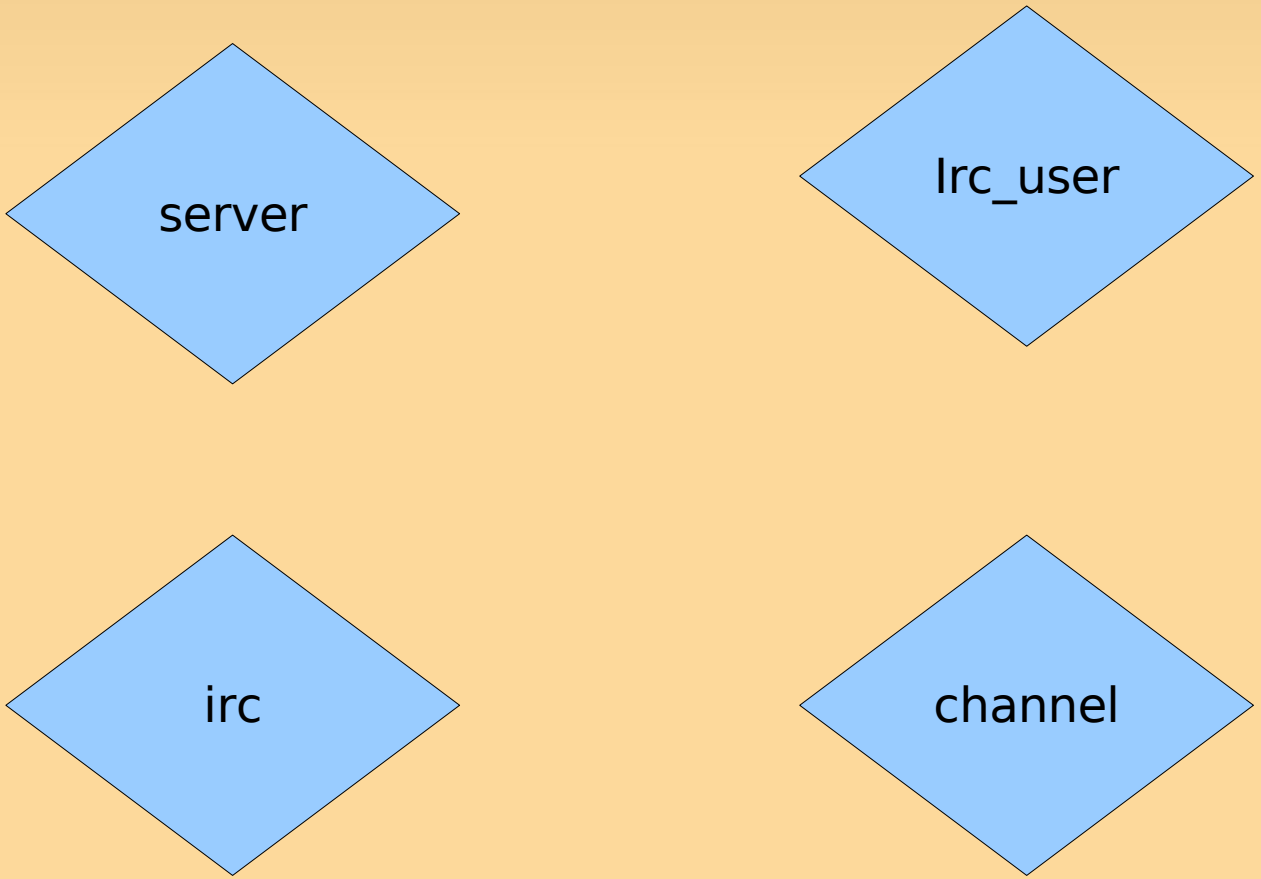
Serveur IRC

- Objectifs :
  - Utilisisation d'ERLANG
  - serveur IRC basique :
    - Discutions
    - Channels

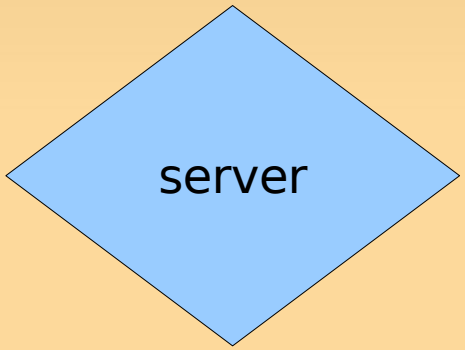
Serveur IRC

- Architectures des modules :



Serveur IRC

- Architectures des modules :



un gen_server qui connait tous les utilisateurs et tous les channels.

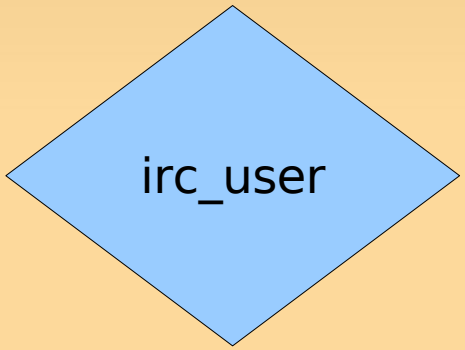
```
-record(state, {users, channels}).
```

users = liste de {Pid, Nick}

Channels = liste de {Pid, Channel}

Serveur IRC

- Architectures des modules :



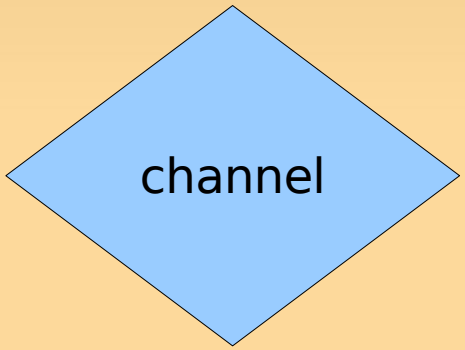
un gen_server qui controle le socket, il y en a un par utilisateur.

C'est ce processus qui reçoit les messages et les interprète,

il sert aussi à envoyer les messages.

Serveur IRC

- Architectures des modules :



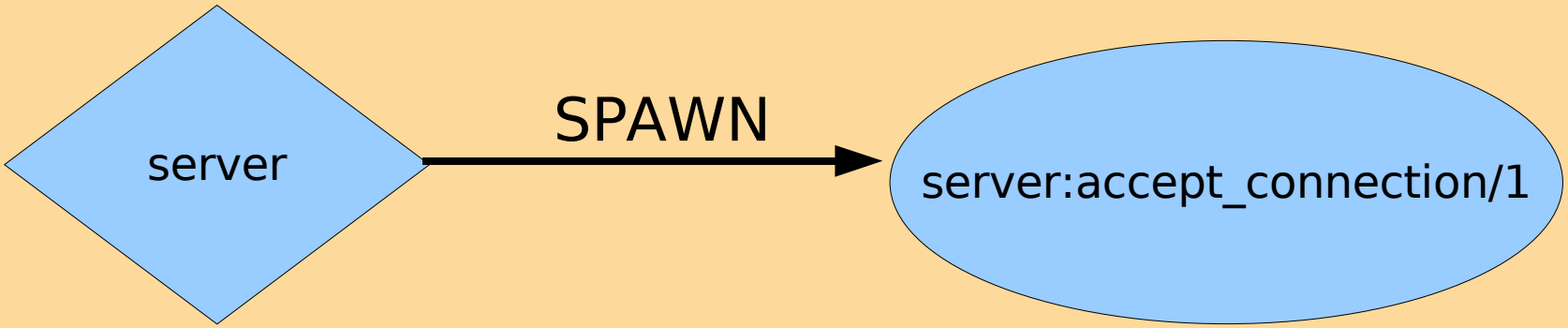
un gen_server qui gère un channel. Il connnait la liste des utilisateurs connectés

```
-record(state, {c_name, topic, users}).
```

Serveur IRC

- Lancement du serveur :

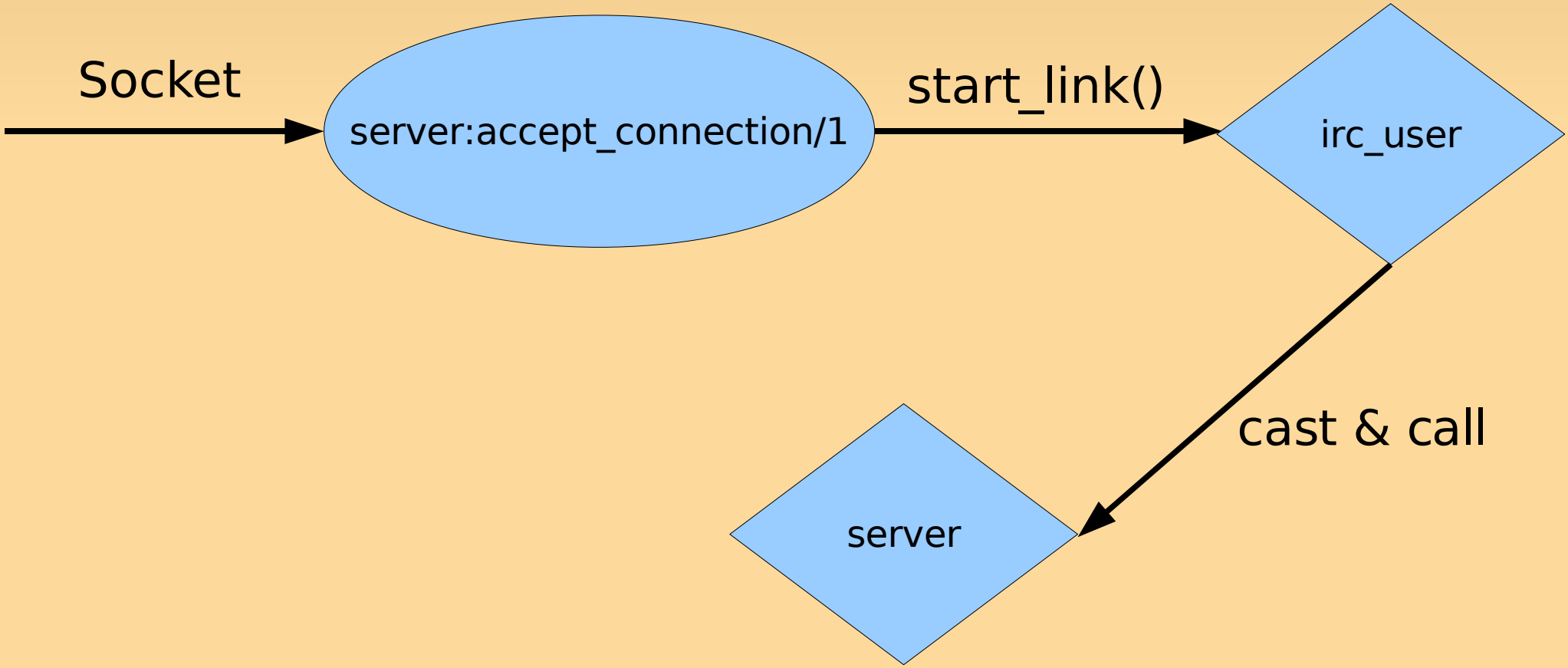
server:start_link().



Ce processus accepte les connections et lance un proccesus irc_user

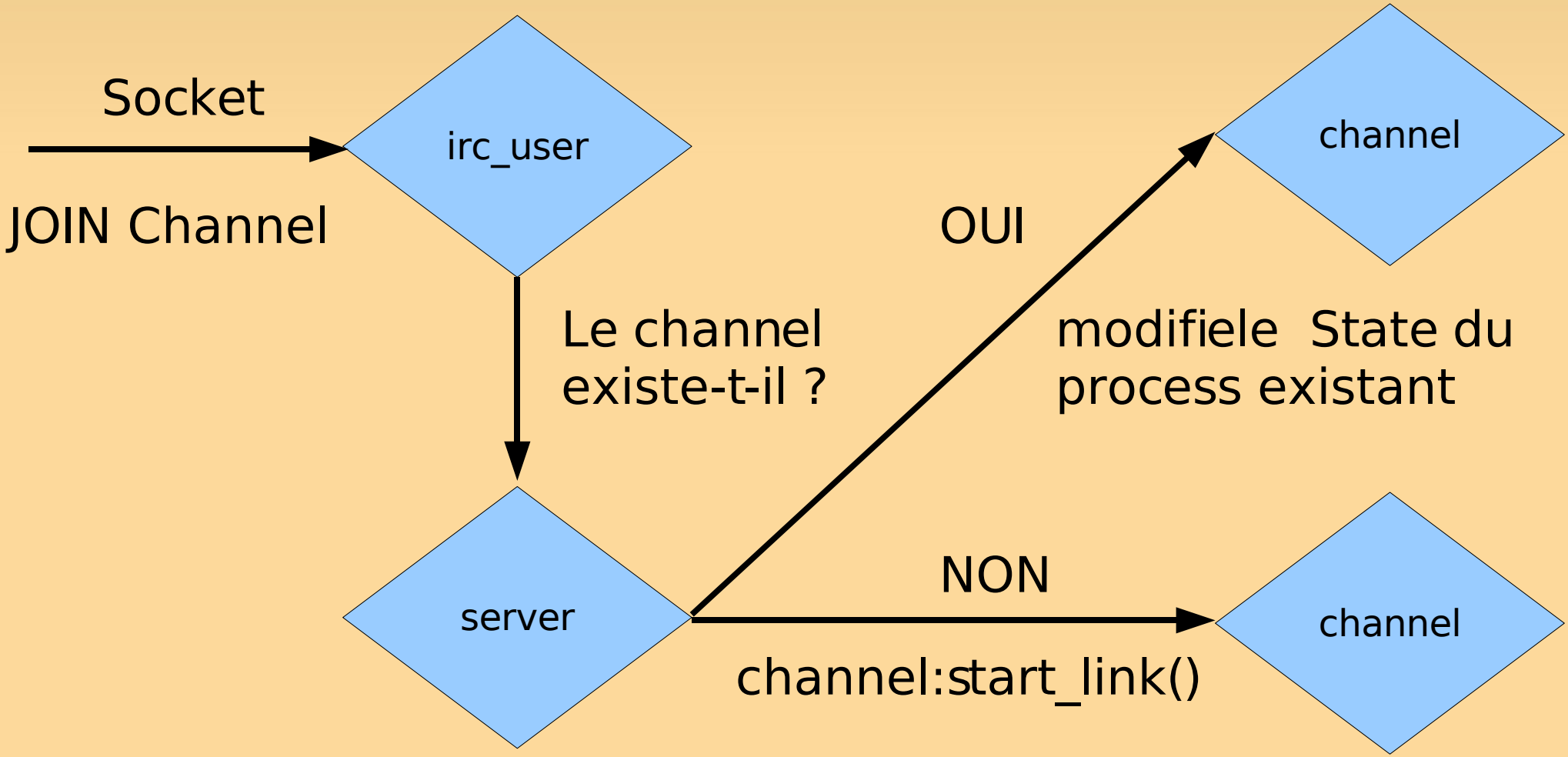
Serveur IRC

- Connections :



Serveur IRC

- Gestion des channels :



Serveur IRC

- Conclusions :

  - Erlang : c'est top !

  - IRC : rfc 2812 pas forcément facile à respecter.

  - client testé :
    - Xchat: aucun bug détecté
    - Gaim : personnes présentes sur le chan avant l'entrée ne sont pas affichées
    - mIRC : aucun bug détecté