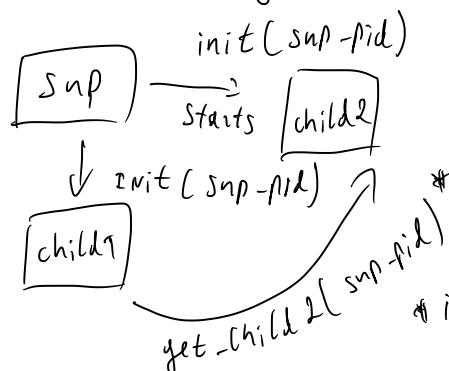


Sibling discovery in Supervisors - the otp usage patterns

- 0) Global context container
- Erlang names
 - Application ENV
 - DI container
- 1.) Register globally unique names with prefix
- 2.) Inject using which method



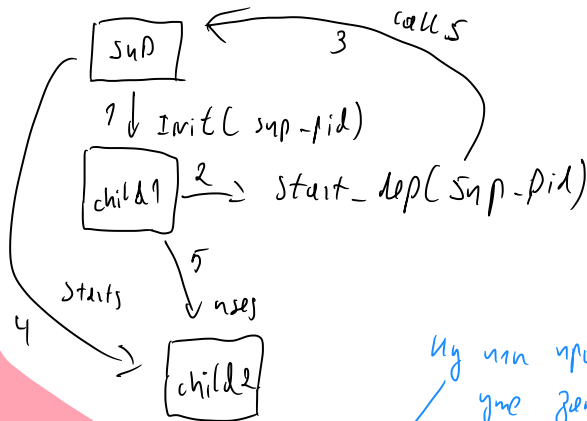
Todo:

გამოწილ სერვისებს უნდა ჰქონდეთ
საპროცესო იდენტიფიკატორი

* iterates using which - children

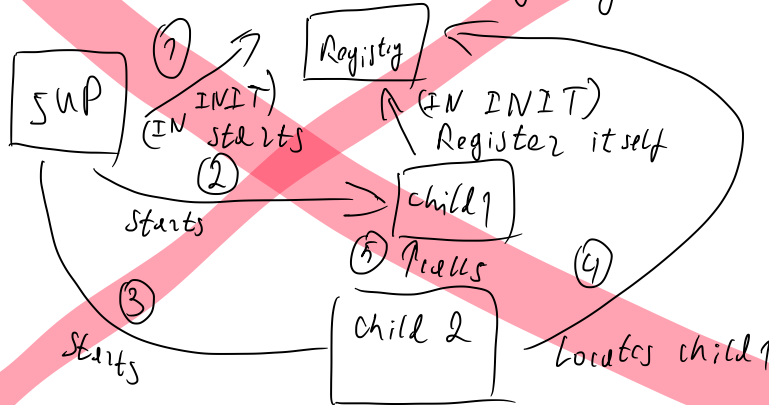
- 3.) Start up on demand

3)



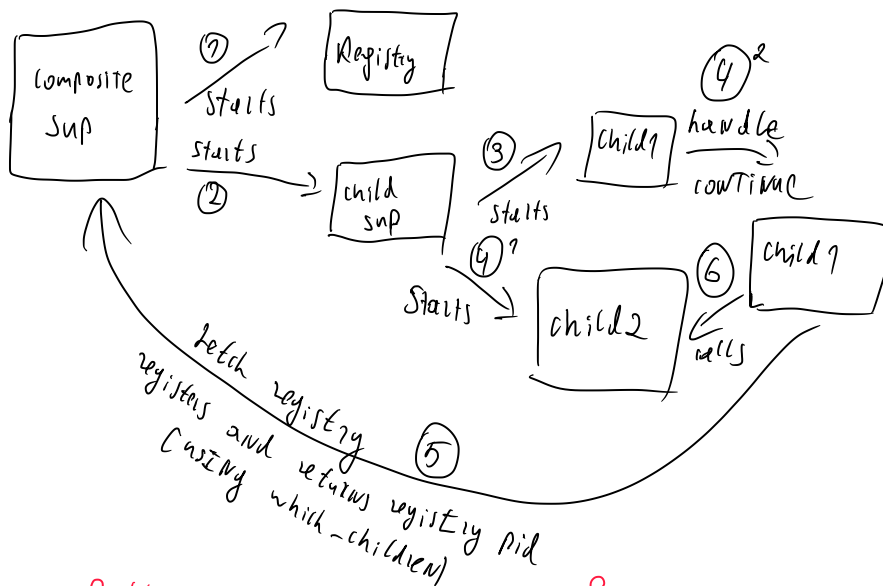
by now you're supposed to have registered Registry

4) IN sup INIT start registry



Composite game ↓

5) Composite Supervisor



Problems with steps sync?

just do use steps IN INIT or handle

just do use `sup` `init` or `handle_continue`
Do NOT save `pid`! Always `fetch`
If you save `pid` \rightarrow `LINK`

Conclusion: OTP gets from no ex much more job
programm `reorganizing` `knows`.

OTP is much more than just a straightforward
usage of `GenServers` and occasionally couple of supervisors.
I hope that elixir community will move beyond
just digging into obvious advantages that `otp`
elements give and will form some higher level
patterns.

I do believe that `otp` is a more than just
sum of its parts however useful they are

