OTP Workshop

Supervisors

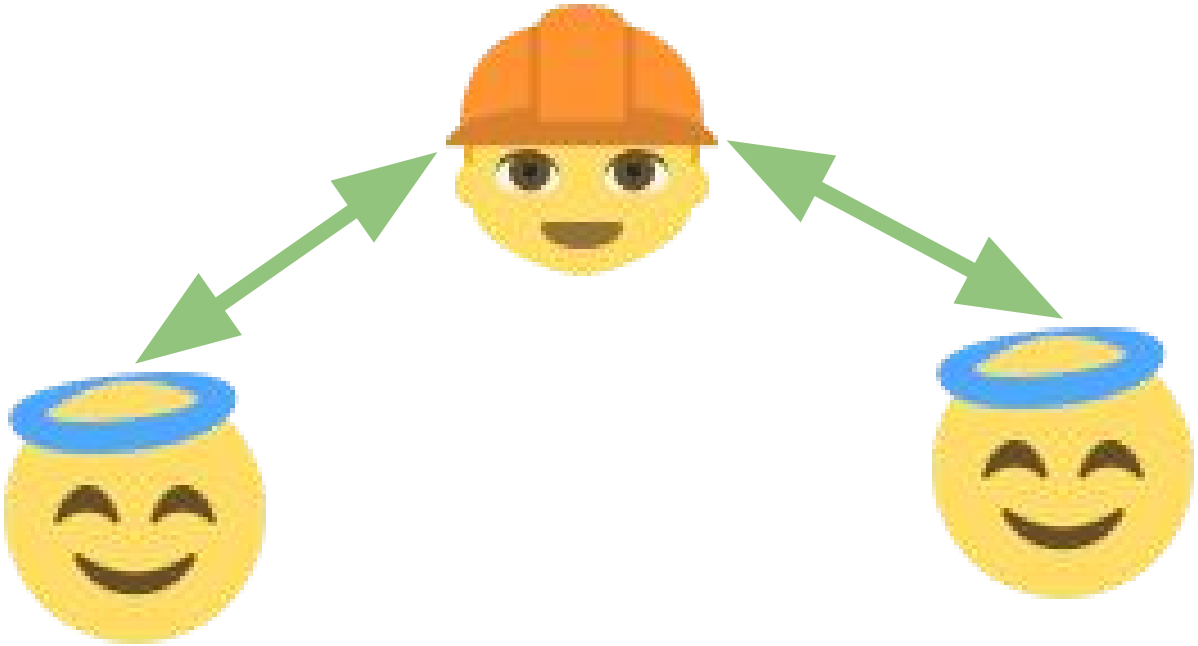
Let it crash again

- ANY process that is restarted may enter BAD state
- Crash ANY process in BAD state
- Restart ANY crashed process to GOOD state
- If restart fails to return to GOOD state then crash AGAIN
- If multiple restarts fail to return to GOOD state then crash supervisor

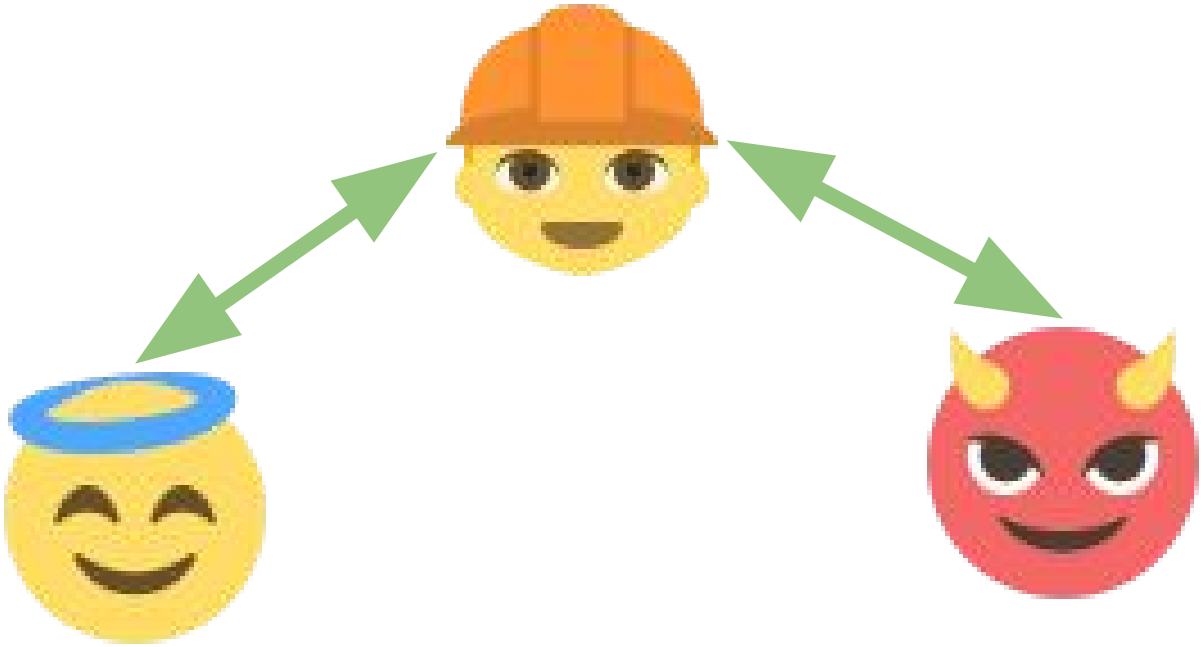
Supervisor

- Supervisor.start_link/2,3
- Supervisor.child_spec/2
- Supervisor.init/2

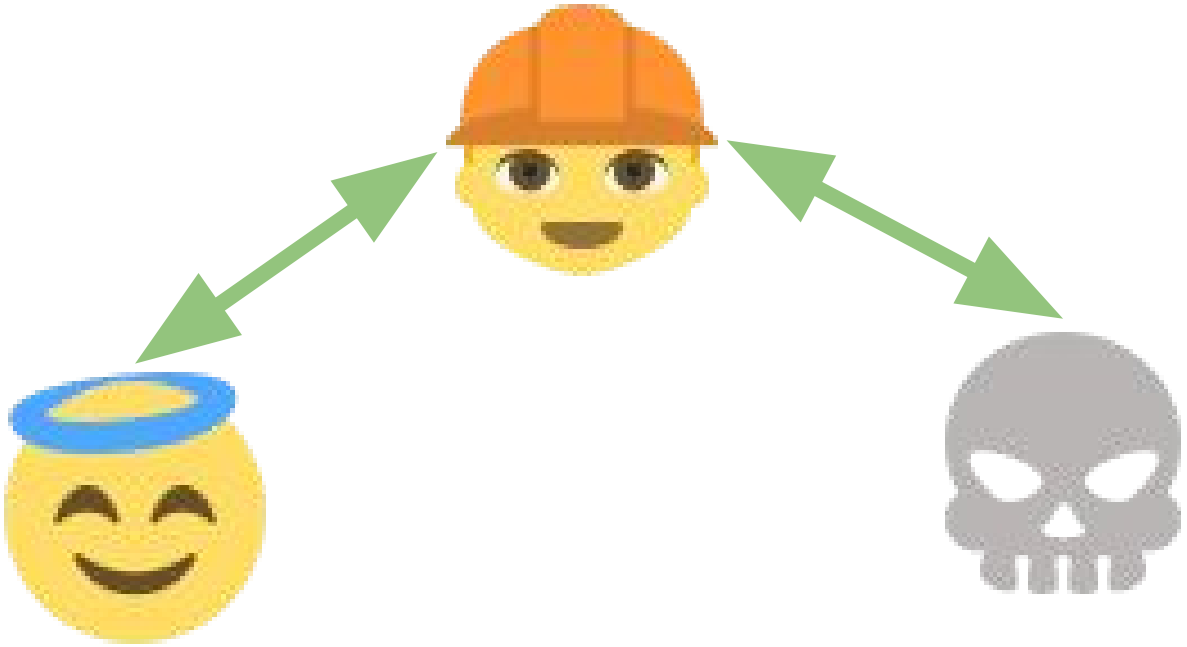
Supervisors



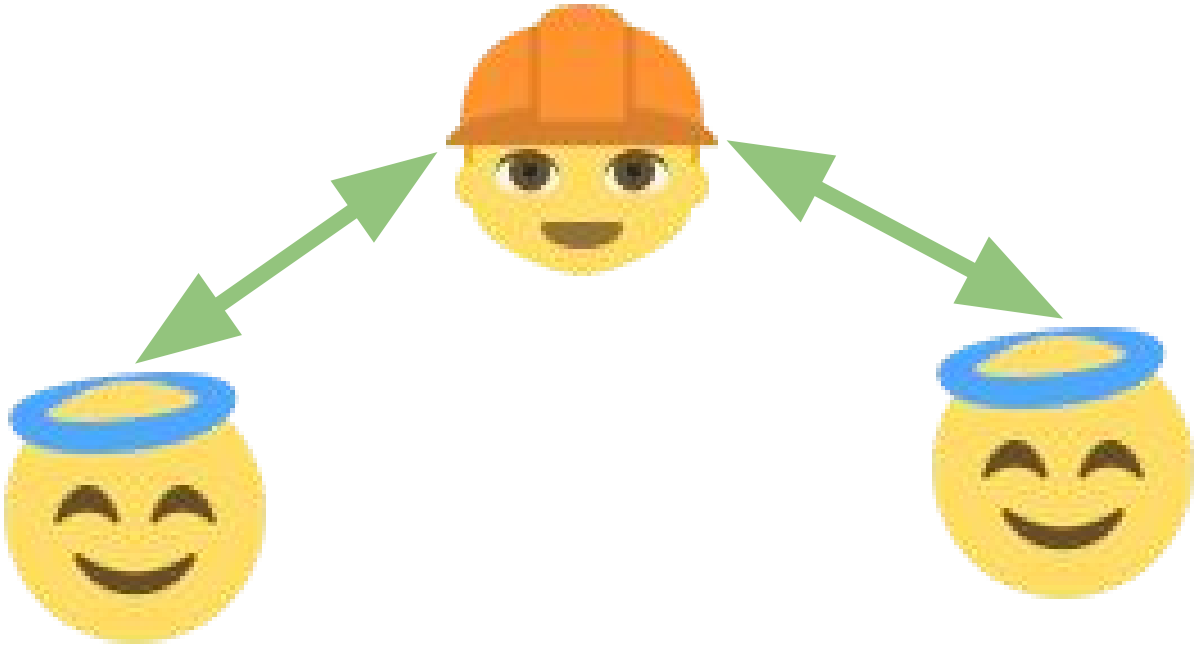
Supervisors



Supervisors



Supervisors



Map Specifications

- Introduced in OTP 18 / Elixir 1.5
- Supervisor.child_spec(module | {module, arg :: term} | map, Keyword.t)
- %{id: id, start: start, restart: restart, shutdown: shutdown, type: type, modules: modules}
- types, modules only used for hot code reloading

Start

- {__MODULE__, :start_link, [arg]}
- use GenServer
- use Supervisor
- use Agent
- use Task

Restart

- :temporary
  - The child process is never restarted
- :transient
  - The child process is restarted only if it exits abnormally; ie with an exit reason that’s not:

    :normal, :shutdown, {:shutdown, term}
- :permanent
  - The child process is always restarted

Child Specification Keys

- :id
  - Uniquely identify a child (defaults to module)
- :shutdown
  - timeout | :brutal_kill
- :type
  - :worker | :supervisor
- :modules
  - :dynamic | [module]

Intensity

- :max_restarts
  - Defaults to 3
- :max_seconds
  - Defaults to 5
- Supervisor shuts down if more than max restarts in max second
- Supervisor.init(children, [max_restarts: restarts, max_seconds: seconds])

Child Communication

- Children started in order and terminated in reverse order
- Younger children initiate communication with older children
- Younger children call (sync) or cast (async) to older siblings
- Older children reply (async) or cast (async) to younger children

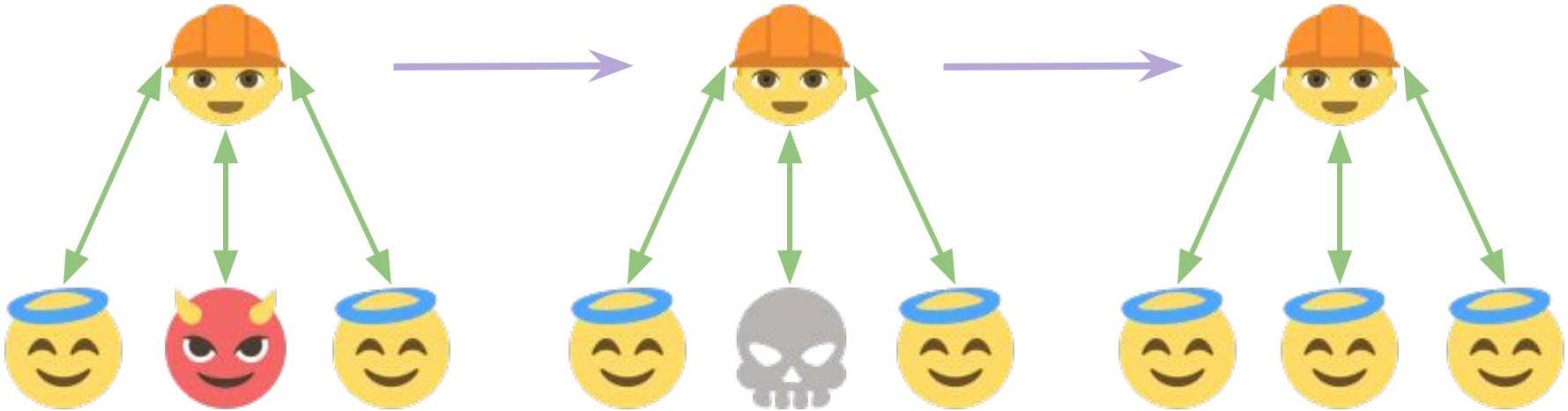
Strategies

- :one_for_one
- :rest_for_one
- :one_for_all
- :simple_one_for_one

One for one

- Children are independent
- Errors are isolated
- When child restarted older siblings may not be alive
- Children usually cast (async) or do not communicate with siblings

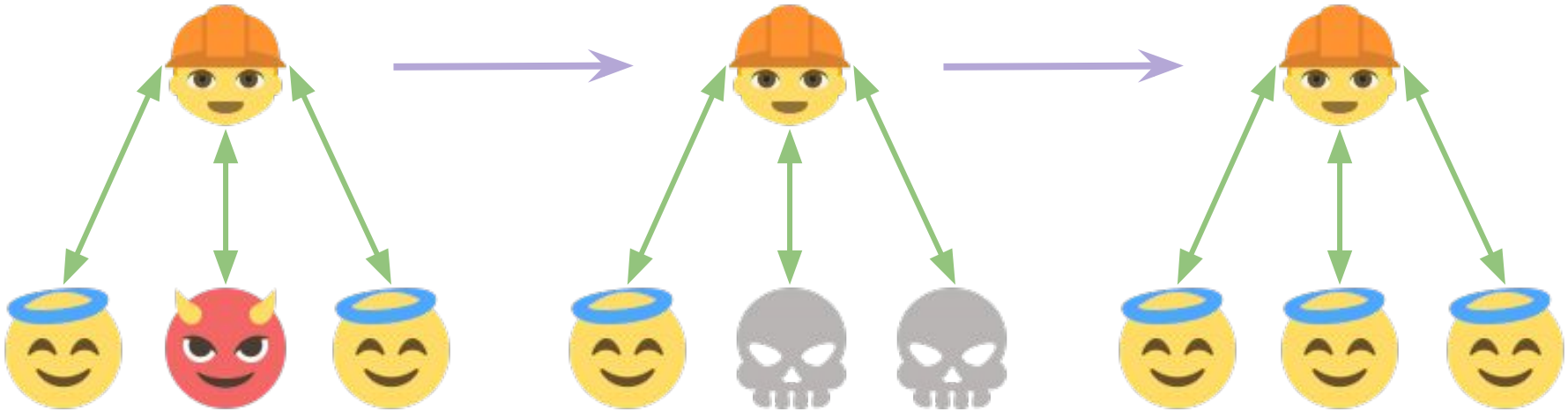
One for one



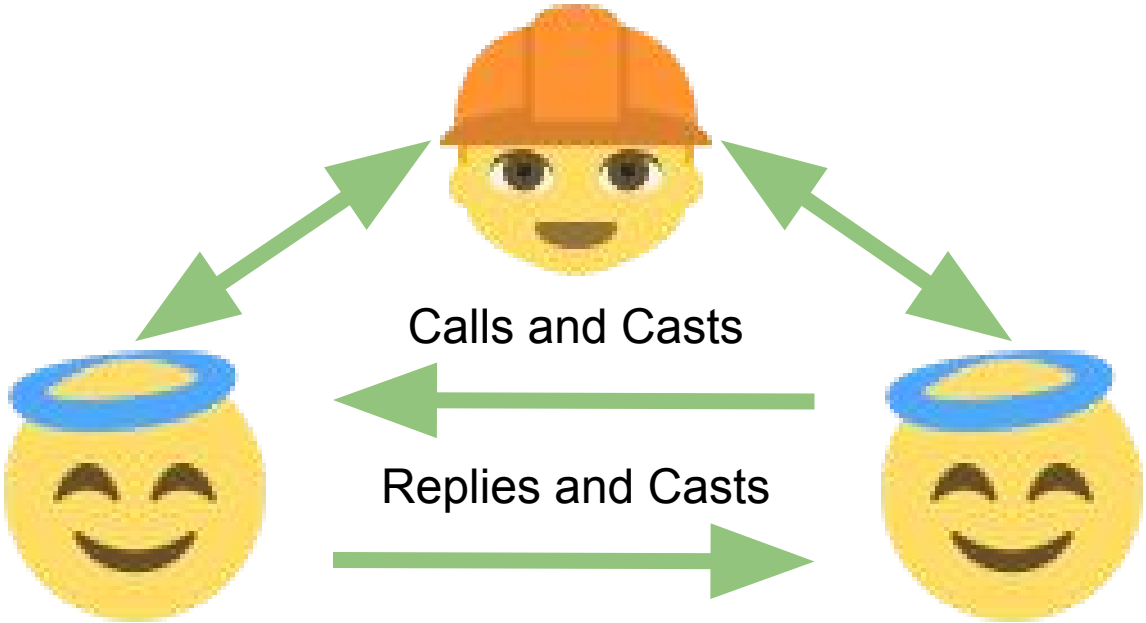
Rest for one

- Children are dependent on younger siblings
- Errors are propagated to all younger siblings
- When child restarted older siblings will be alive
- Younger children usually call (sync) older siblings

Rest for one



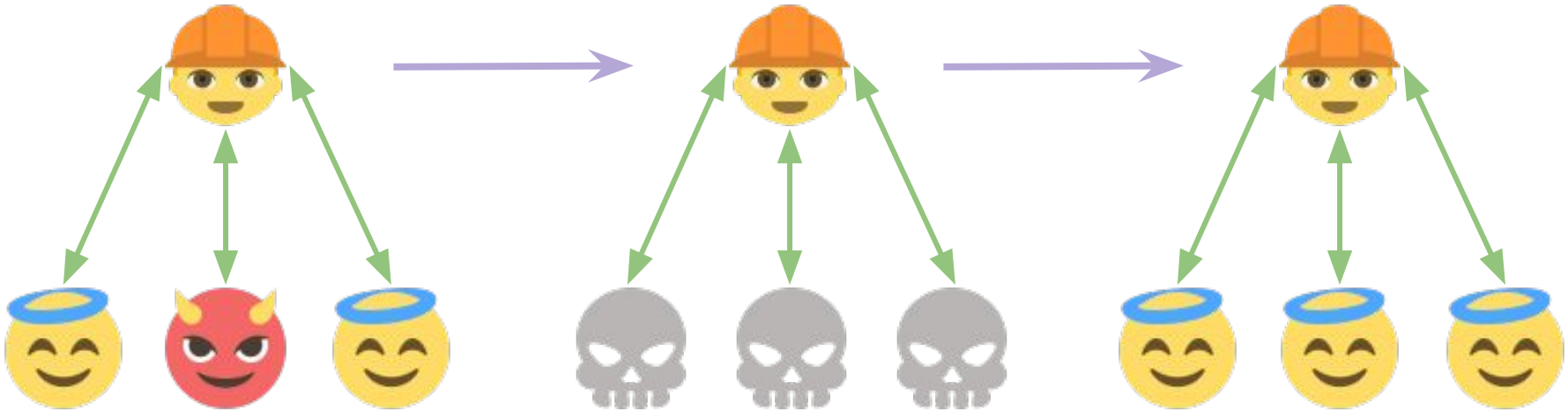
Rest for one



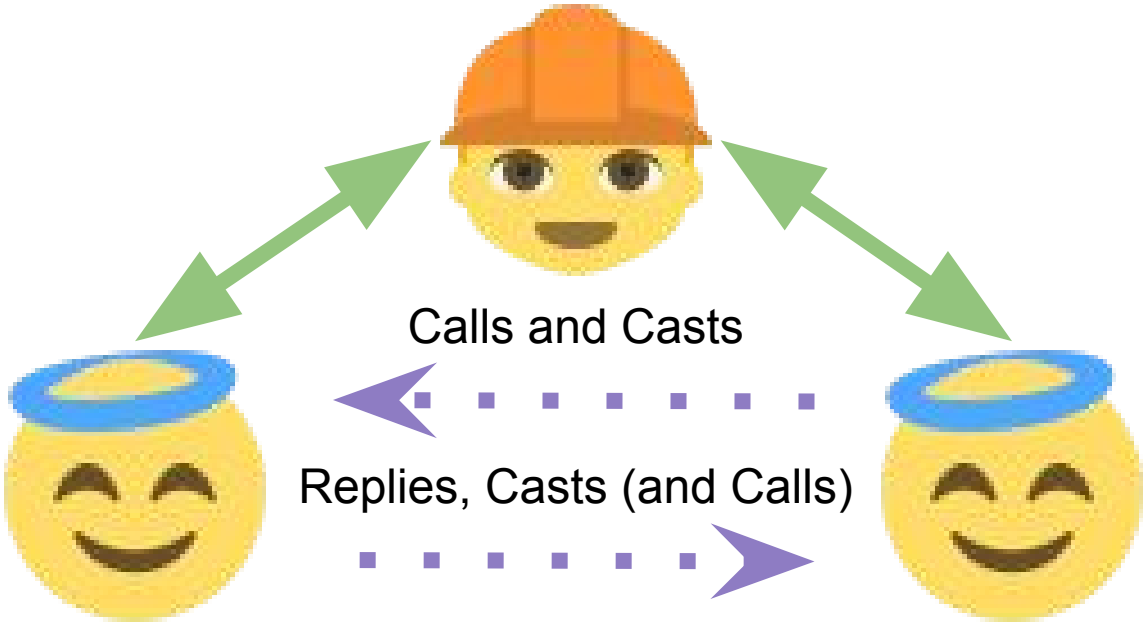
One for all

- Children are dependent on younger or older siblings
- Errors are propagated to all siblings
- When child restarted older siblings will be alive
- Younger children initiate communication with older siblings
- Older children enter BAD state when younger sibling crashes

One for all



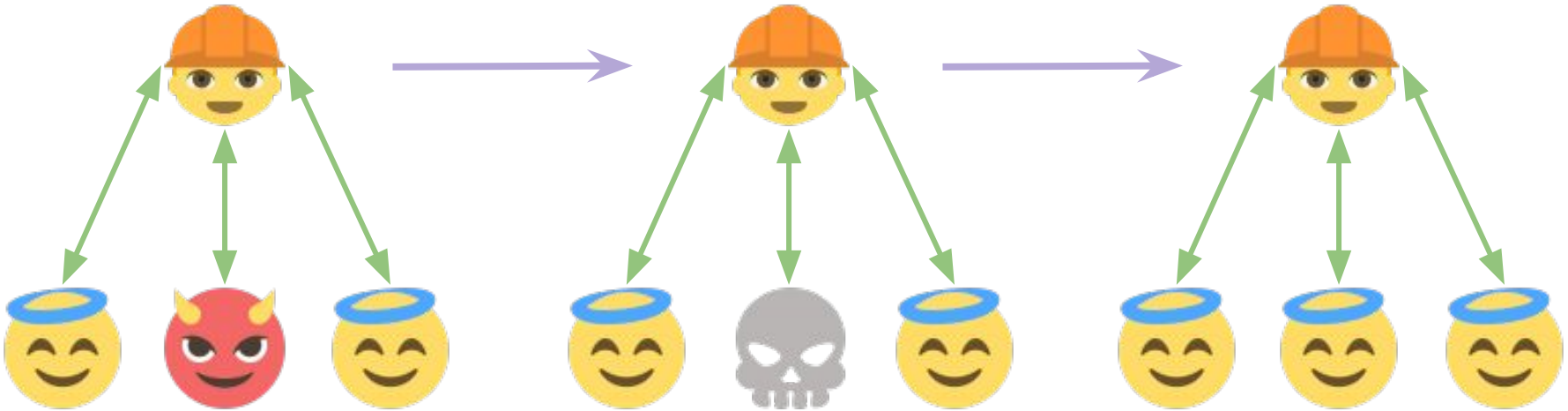
One for all



Simple one for one

- Not simple
- Children are independent
- Errors are isolated
- Children started dynamically
- Children terminated concurrently

Simple one for one



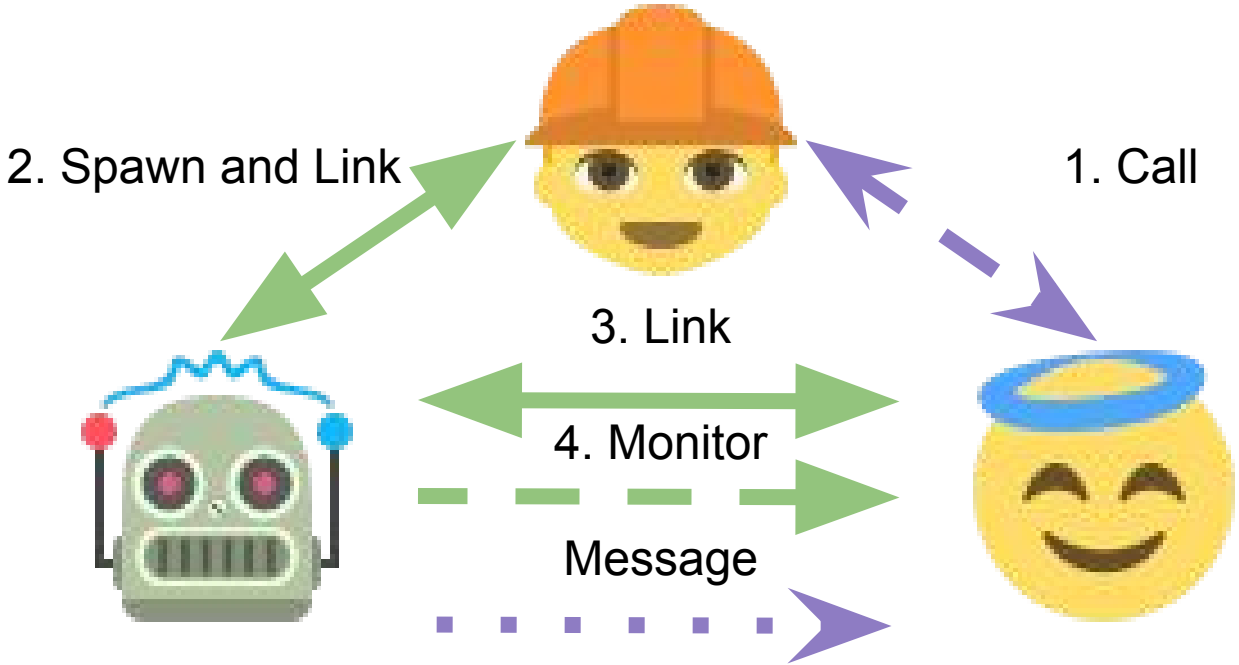
Task.Supervisor

- :simple_one_one for Tasks
- Tasks are independent
- Errors are isolated
- Tasks started dynamically
- Tasks terminated concurrently

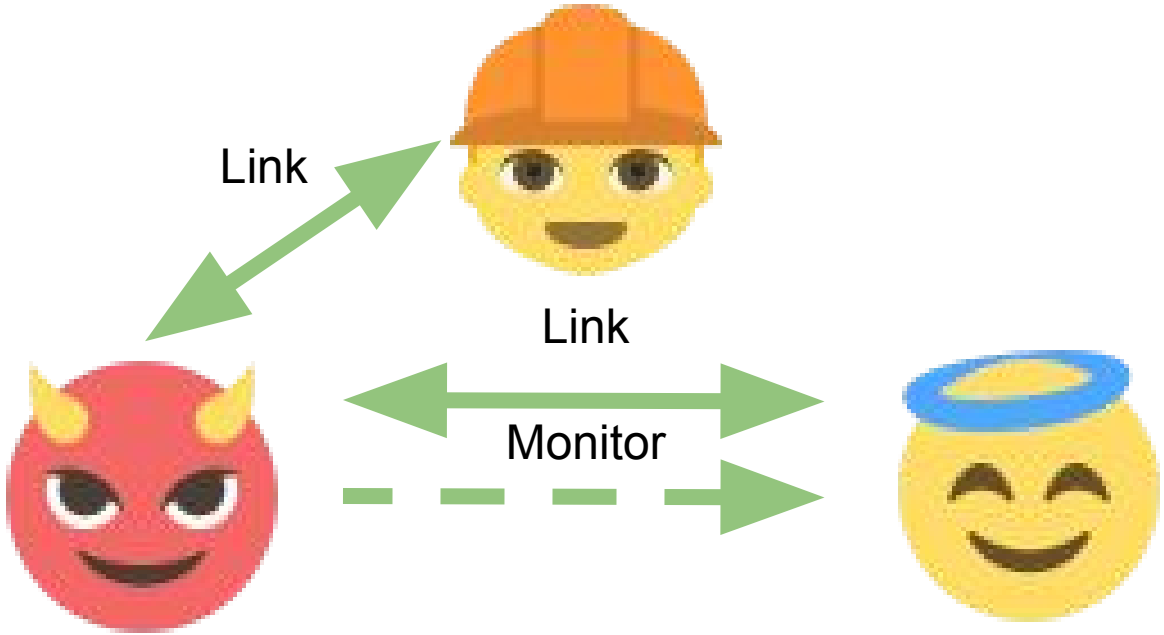
Task.Supervisor

- Task.Supervisor.start_link/1
- Task.Supervisor.start_child/2,4
- Task.Supervisor.async/2,4
- Task.Supervisor.async_nolink/2,4
- Async tasks should use the default restart: :temporary

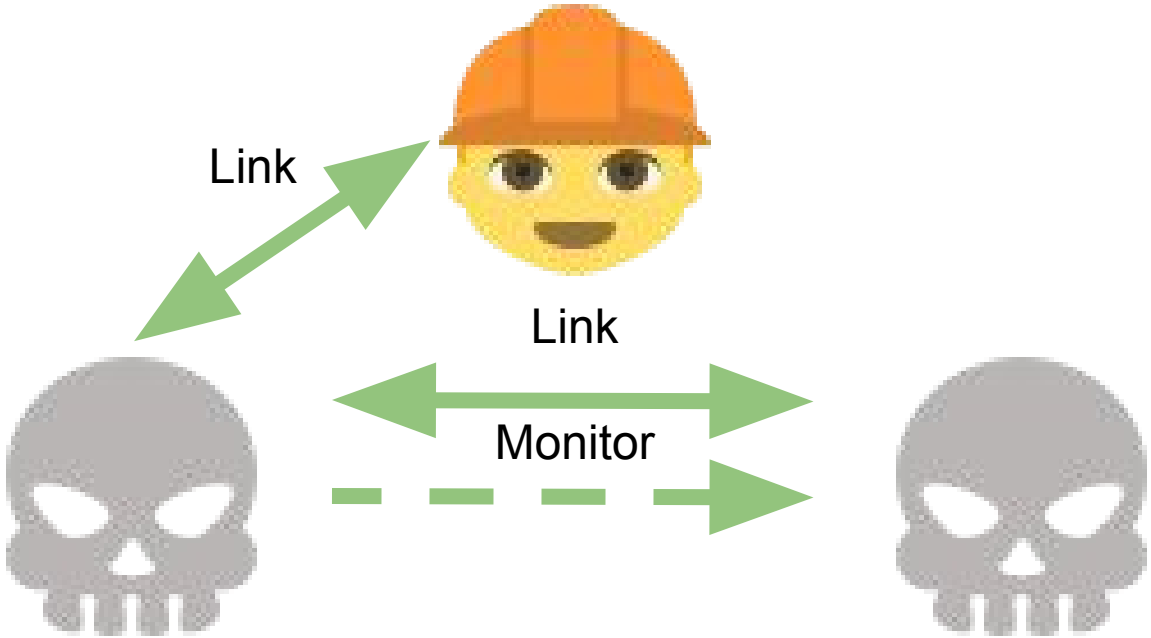
Task.Supervisor.async/2,4



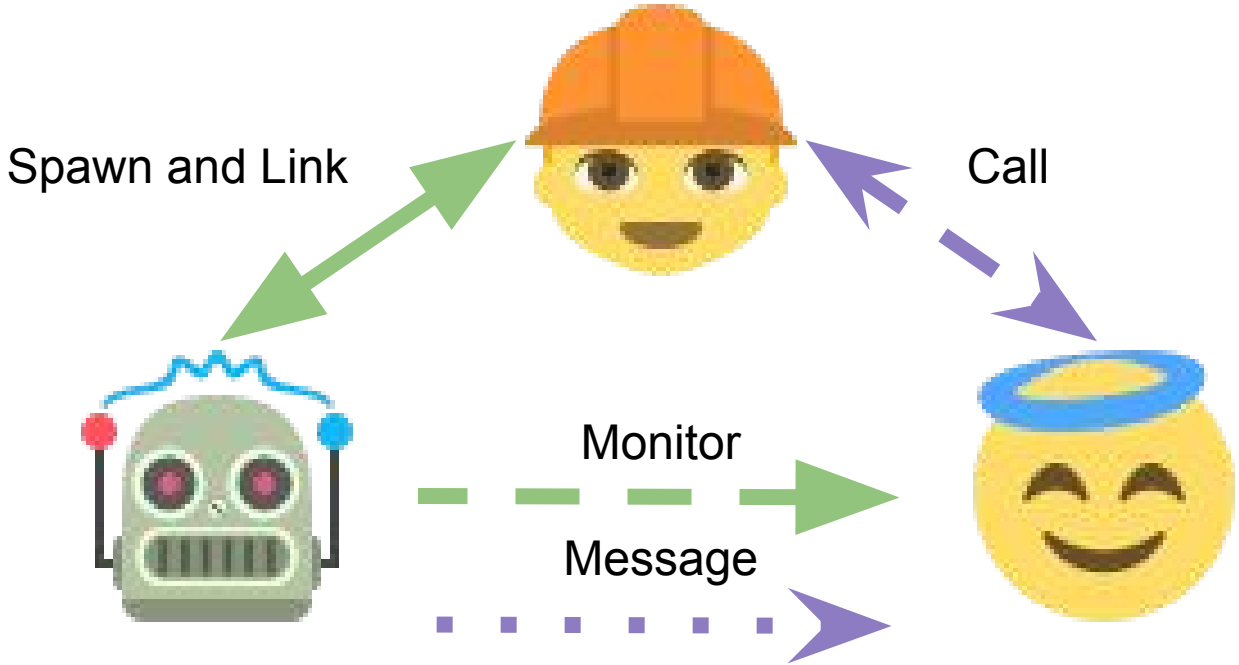
Task.Supervisor.async/2,4



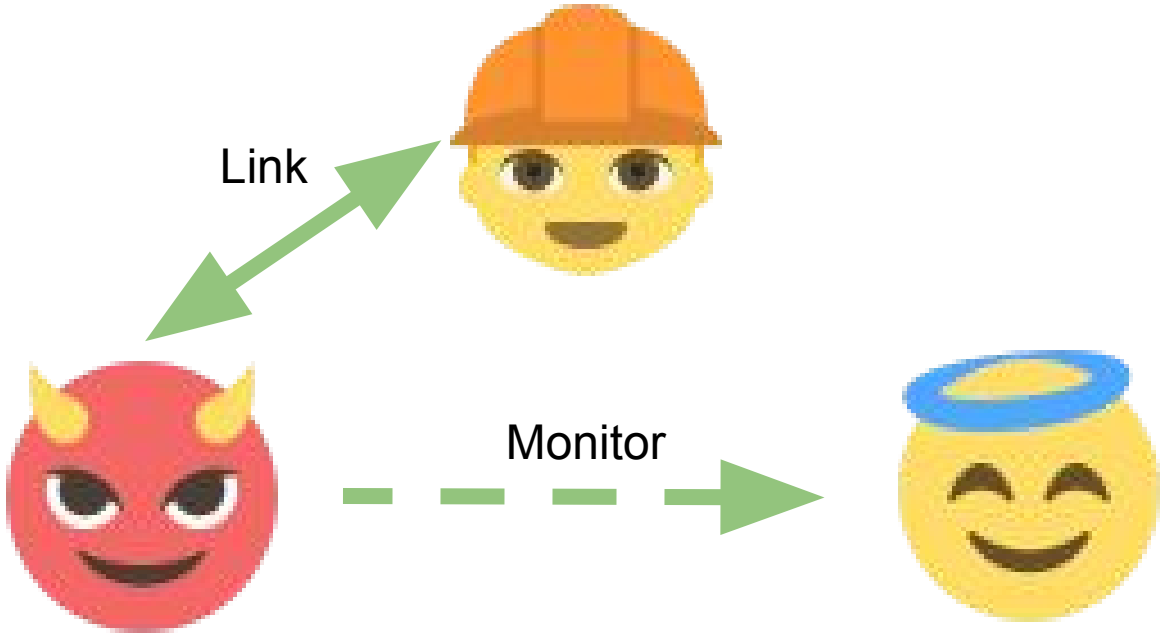
Task.Supervisor.async/2,4



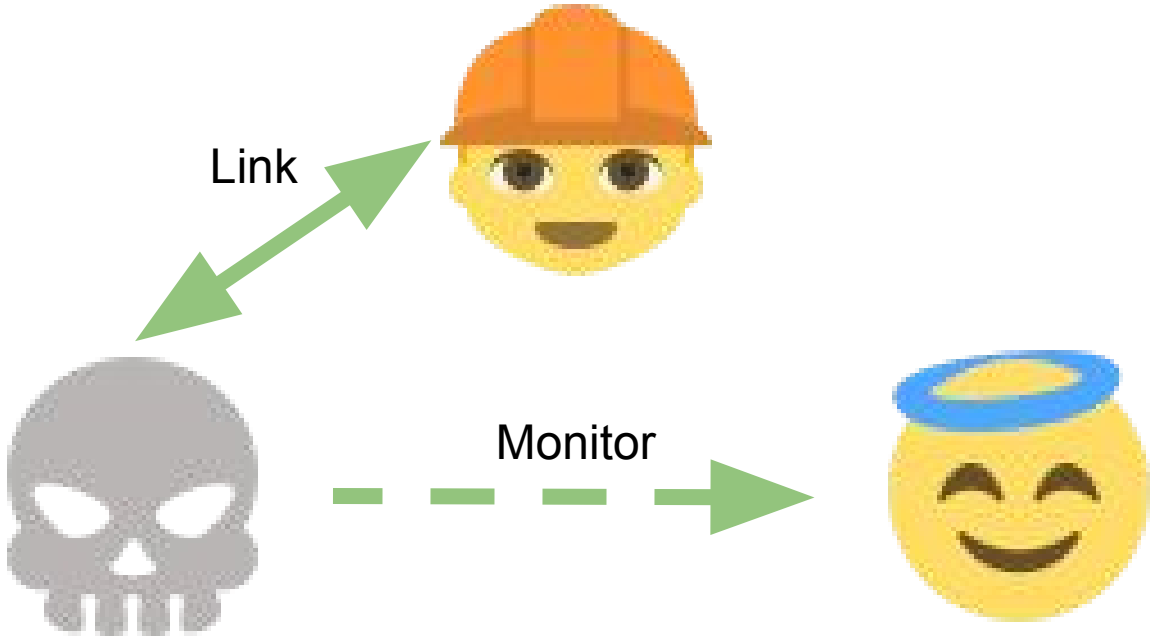
Task.Supervisor.async_nolink/2,4



Task.Supervisor.async_nolink/2,4



Task.Supervisor.async_nolink/2,4



Supervisor problems