

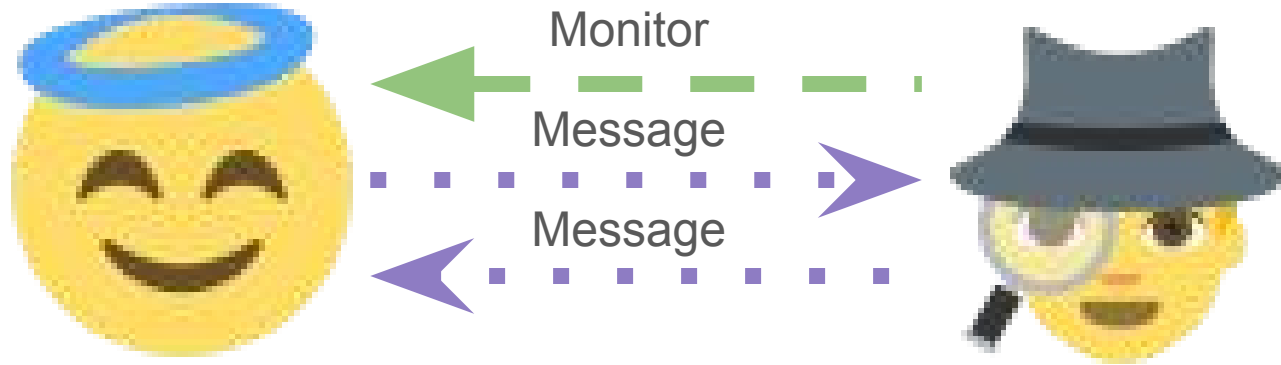
OTP Workshop

Agents and GenServers

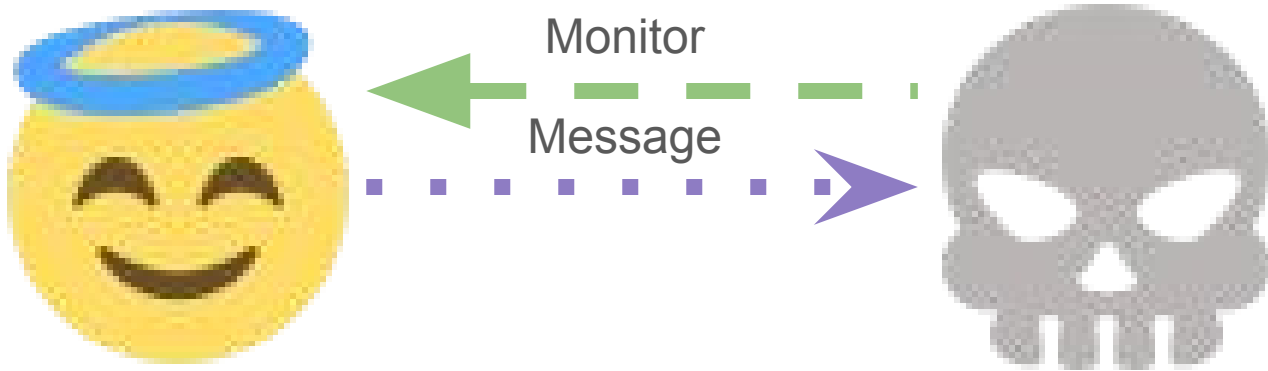
Agent.get/2,3,4,5

- `Agent.get(agent, fn(state) -> ... end, timeout)`
- `Monitor, send, receive, demonitor (with flush)`
- On timeout: `{:timeout, {Agent, :get, [agent, fn(state) -> ... end, timeout]}}`
- On crash: `{reason, {Agent, :get, [agent, fn(state) -> ... end, timeout]}}`

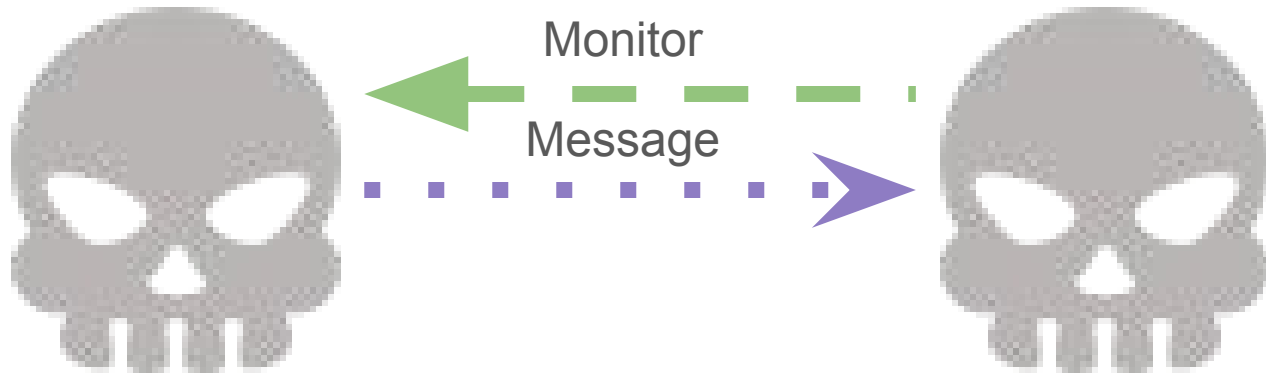
Agent.get/2,3,4,5



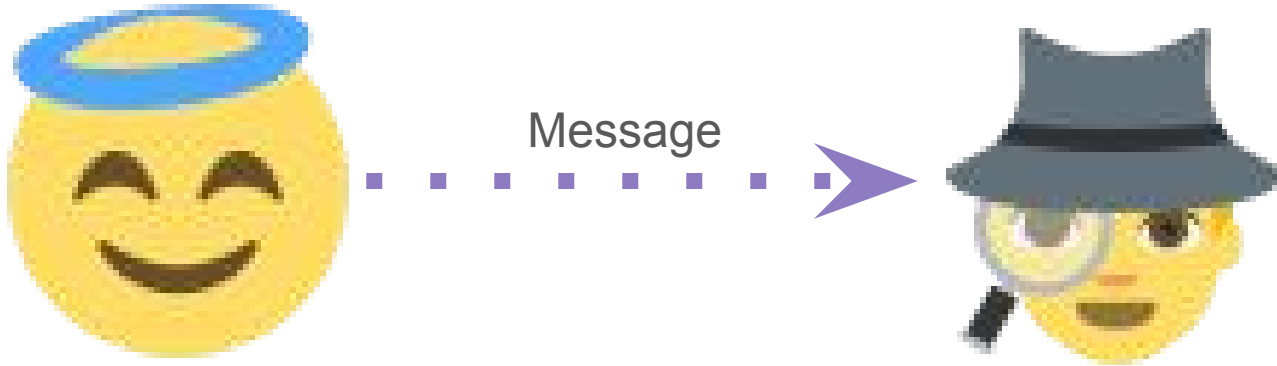
Agent.get/2,3,4,5



Agent.get/2,3,4,5



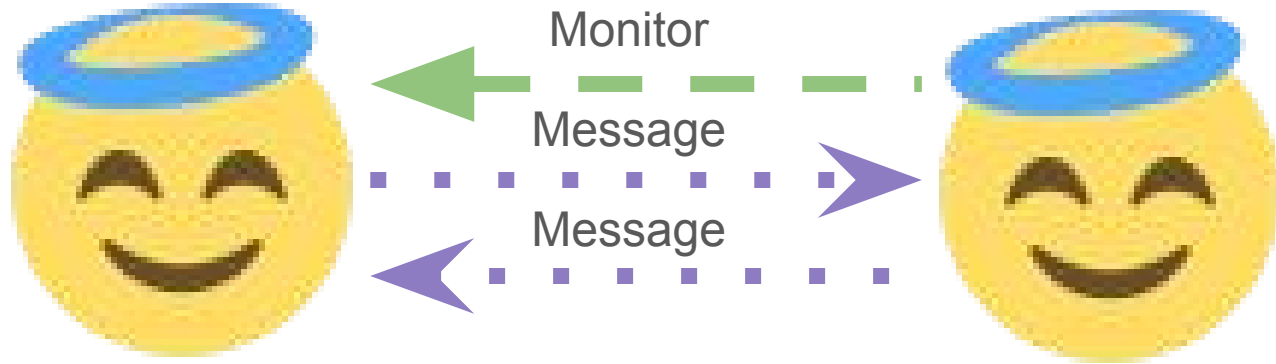
Agent.cast/2,4



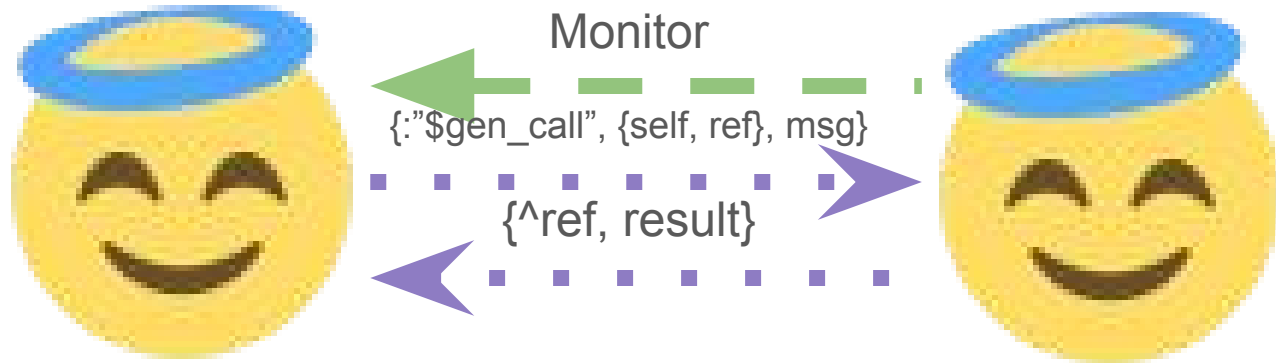
GenServer

- `GenServer.start/2,3`
- `GenServer.start_link/2,3`
- `GenServer.call/2,3`
- `GenServer.cast/2`

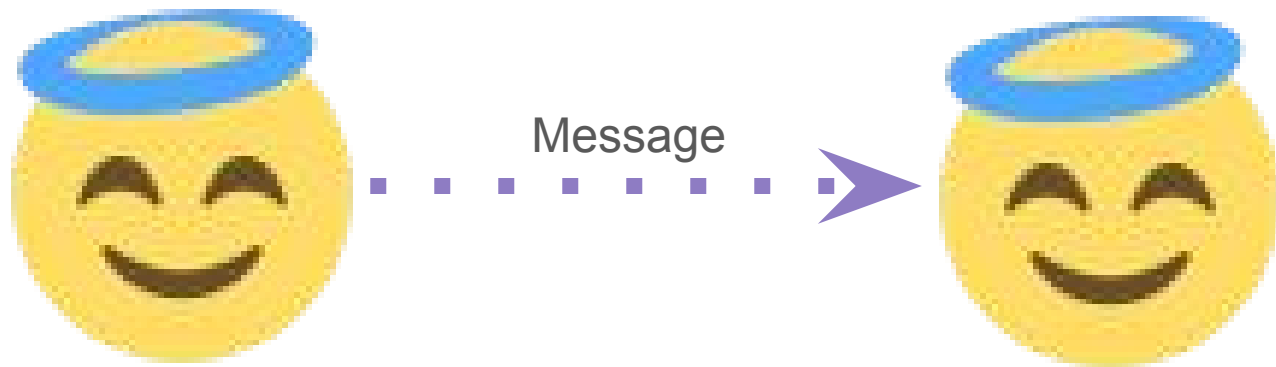
GenServer.call/2,3



GenServer.call/2,3



GenServer.cast/2



How does VM handle process exit?

- Unregister local name
- Cancel timers
- Delete or giveaway ETS
- Send exit signals
- Send monitor signals
- Log termination (if error)

Behaviour Selection

- Task
 - One off functions
 - Shares lifecycle of parent
- Agent
 - Multiple processes to access shared state atomically
- GenServer
 - Long-lived process that handles requests and returns responses

Agent and GenServer problems