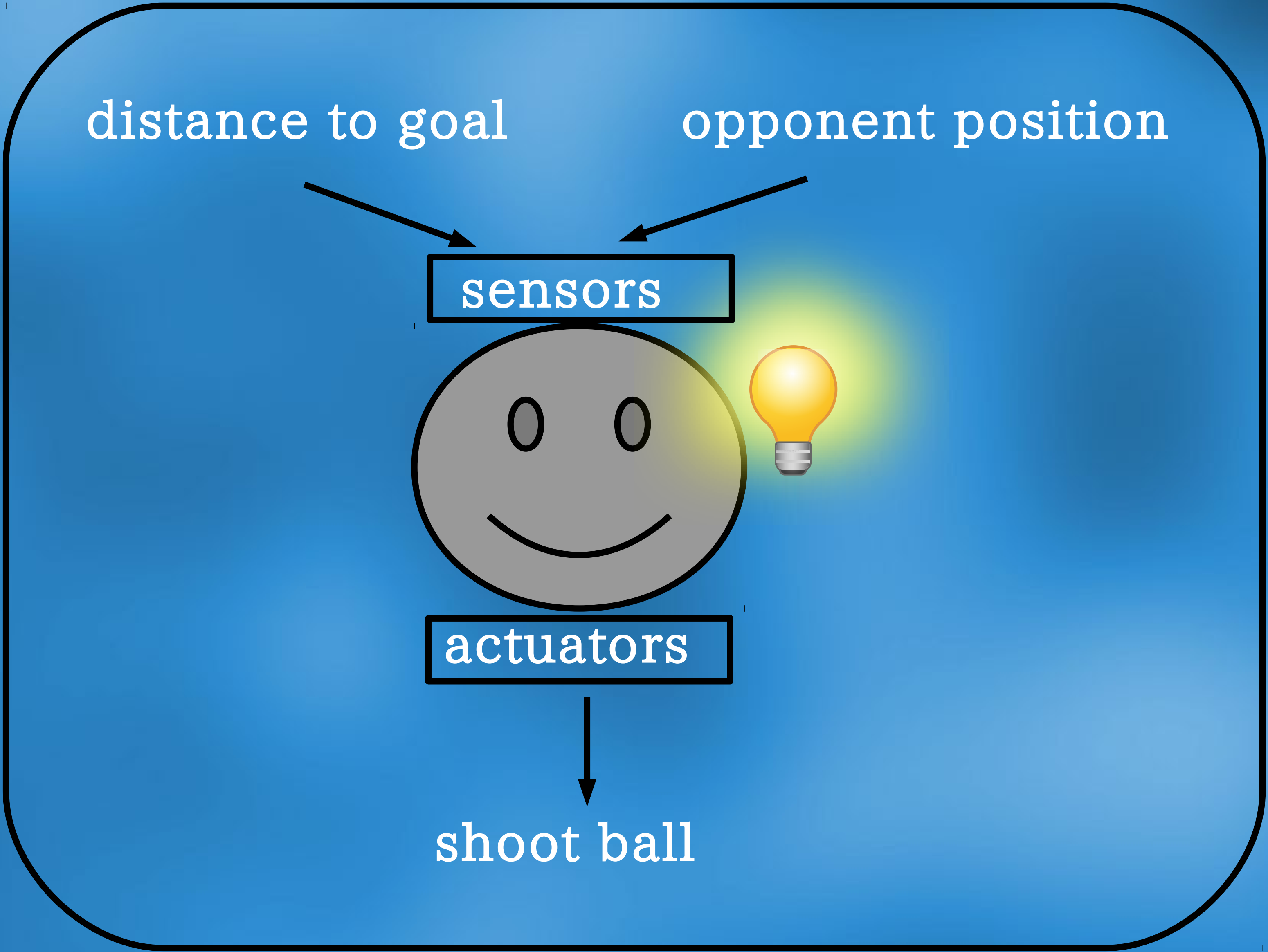
Building Distributed and Robust Multi-agent Systems

Álvaro Fernández Díaz

Clara Benac Earle

Lars-Ake Fredlund

3
5
rthern Bites



distance to goal

opponent position

sensors

actuators

shoot ball

Autonomy

Proactiveness

Reactivity

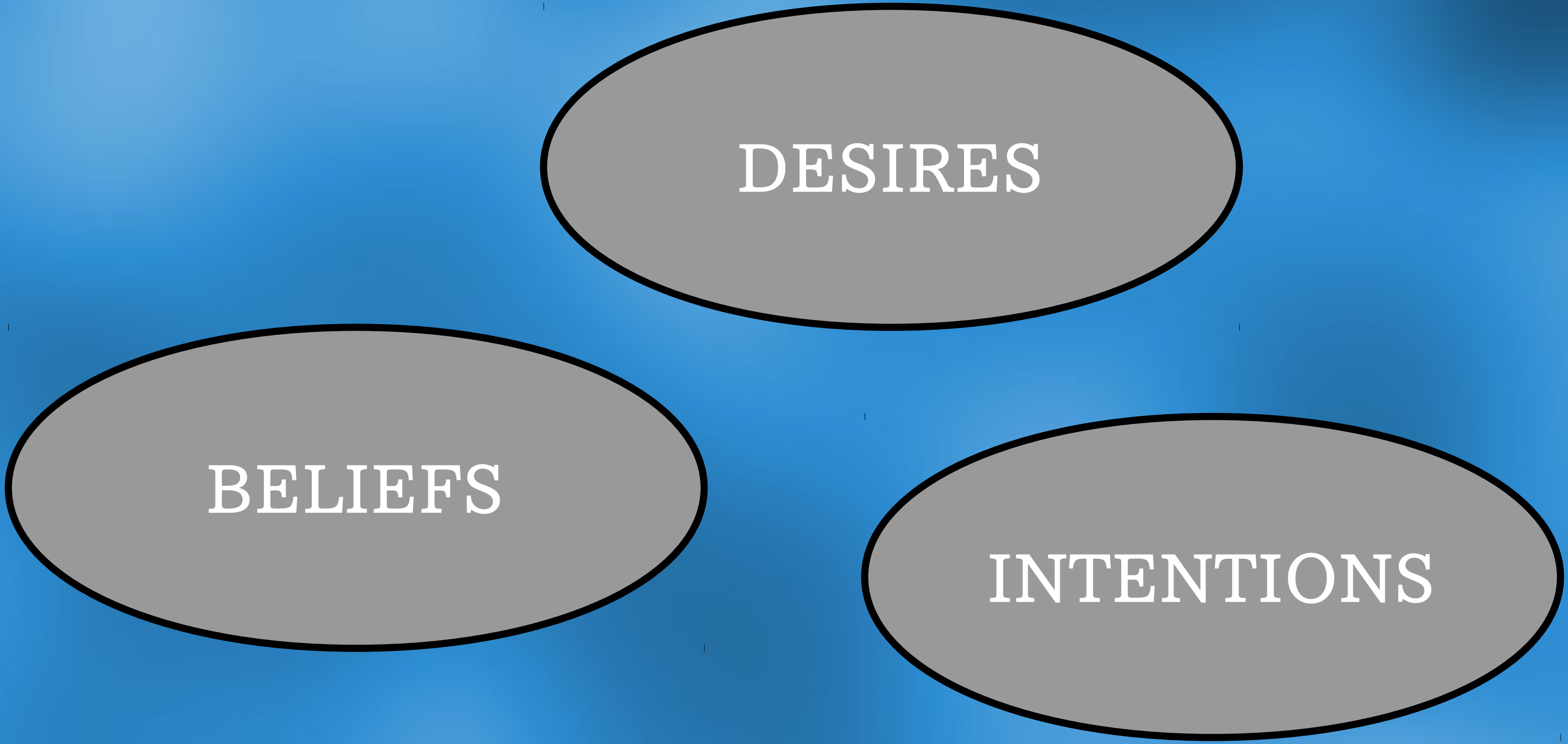
Social ability

.....

A day in an agent's life

A day in an agent's life

Leaves home
Walks to the office
Works
Walks home



DESIRES

BELIEFS

INTENTIONS

BELIEFS

at(home)

DESIRES

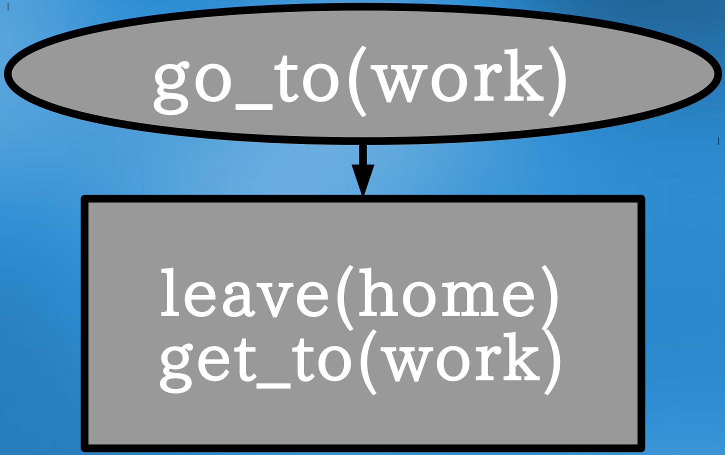
I want to go to work

INTENTIONS

To go to work I need to leave home and walk to the office

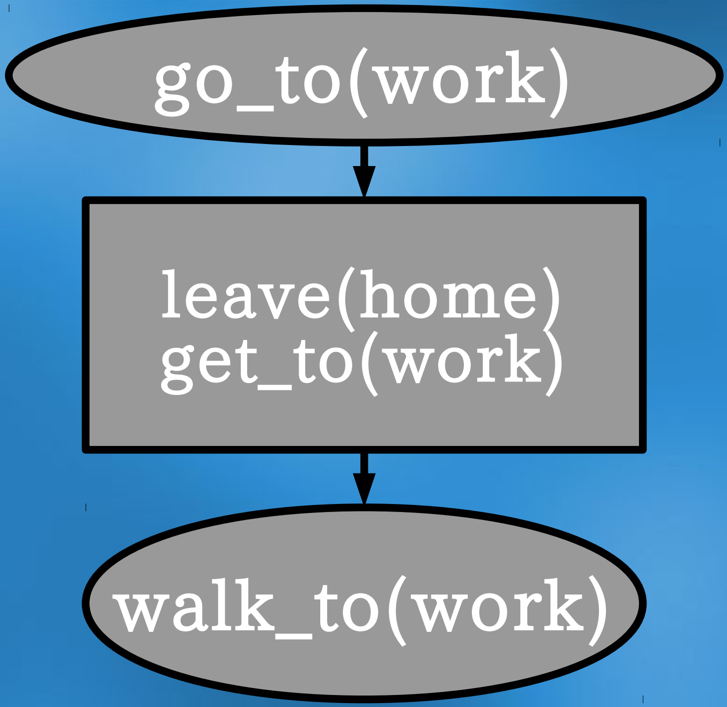
The central idea is that we can architect computer programs as if they have a ‘mental state’

go_to(work)



go_to(work)

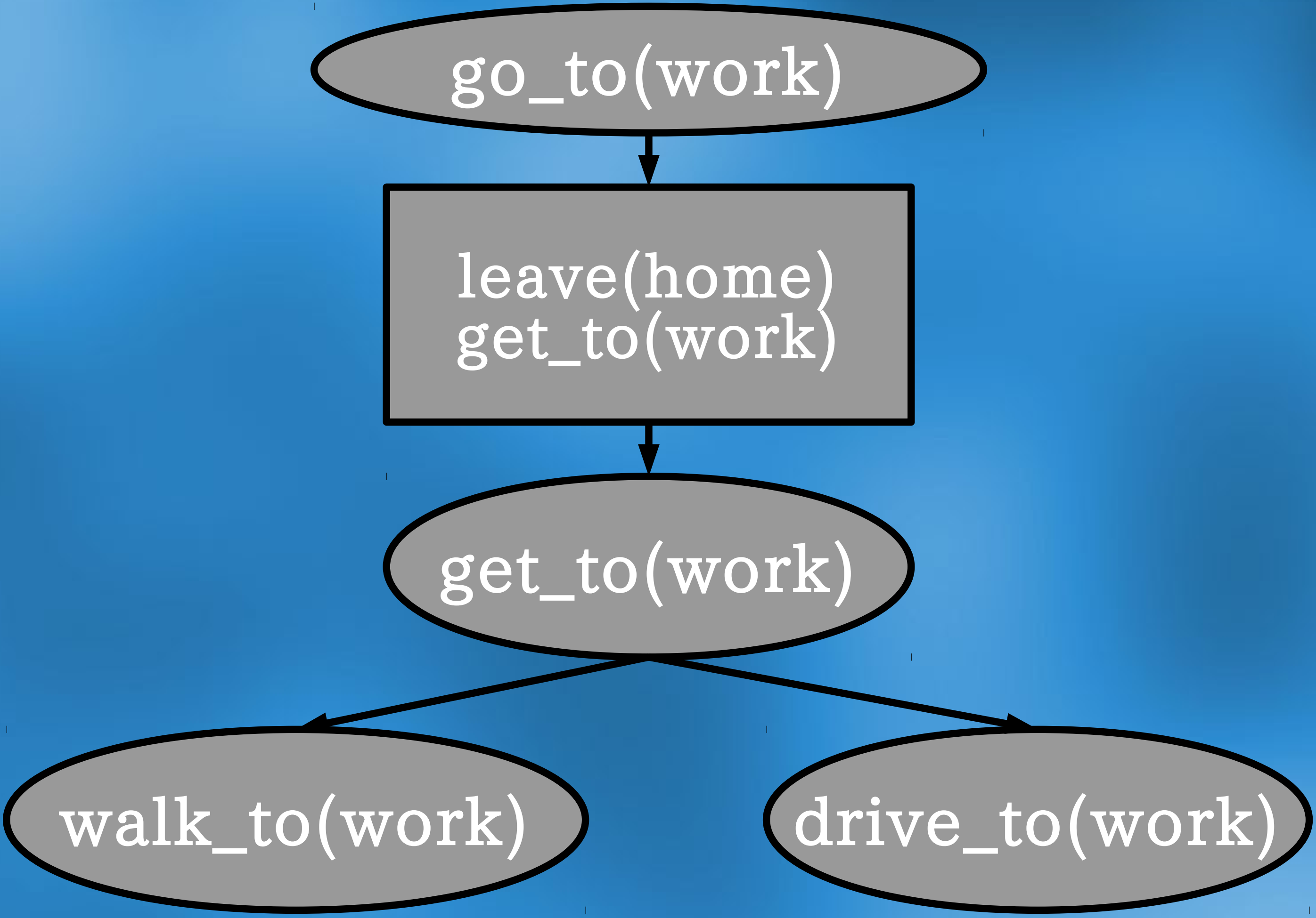
leave(home)
get_to(work)



go_to(work)

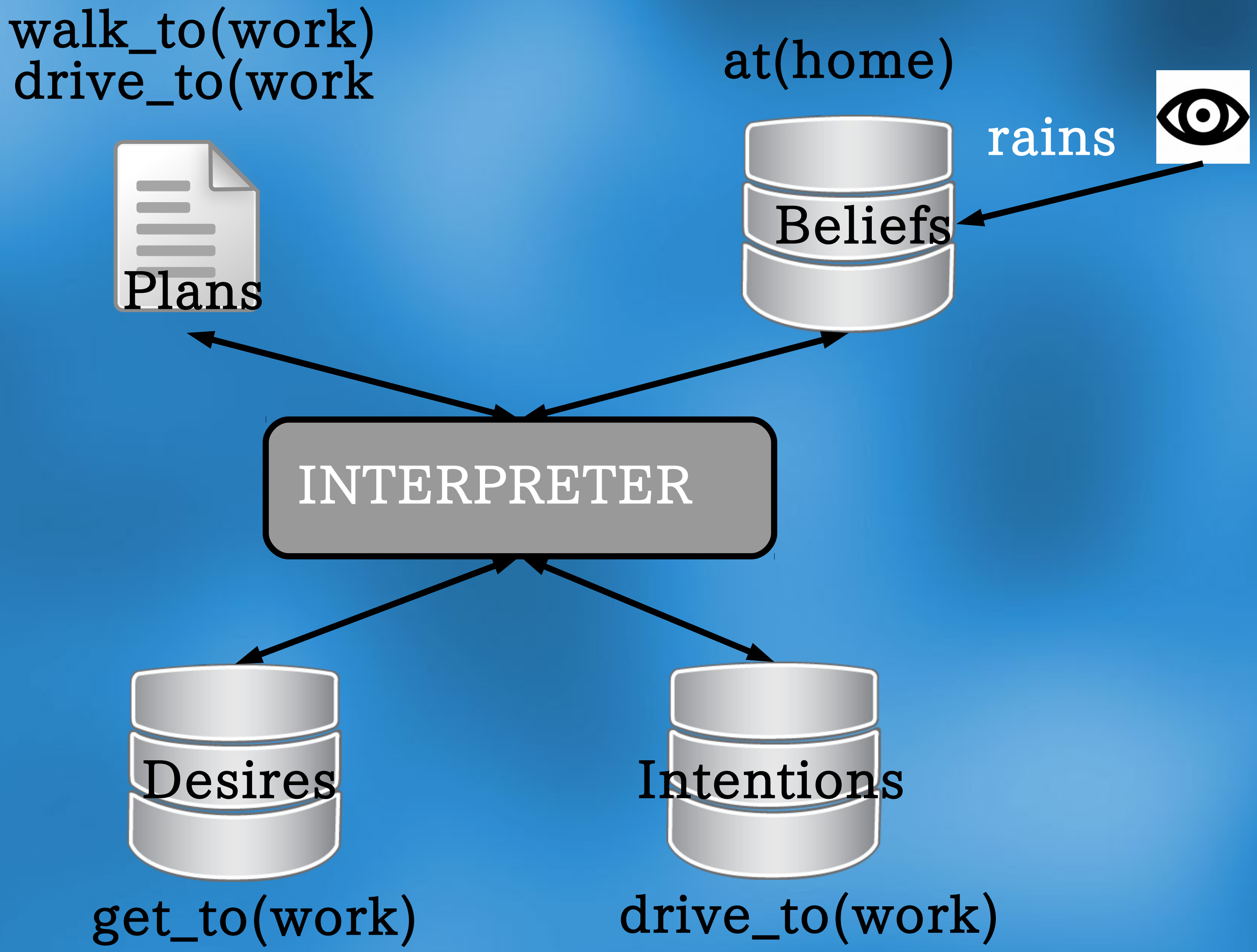
leave(home)
get_to(work)

walk_to(work)


go_to(work)
leave(home)
get_to(work)
get_to(work)
walk_to(work)
drive_to(work)

A day in an agent's life

```
If it rains & has key & car is near
    drives to the office
Otherwise
    walks to the office
```



walk_to(work)
drive_to(work

Plans

at(home)

rains

Beliefs

INTERPRETER

Desires

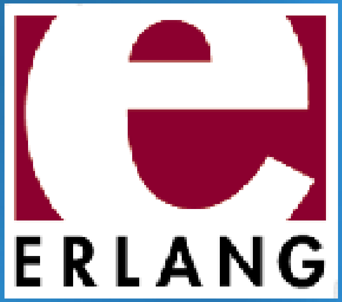
get_to(work)

Intentions

drive_to(work)

How do I program that?

How do I program that?

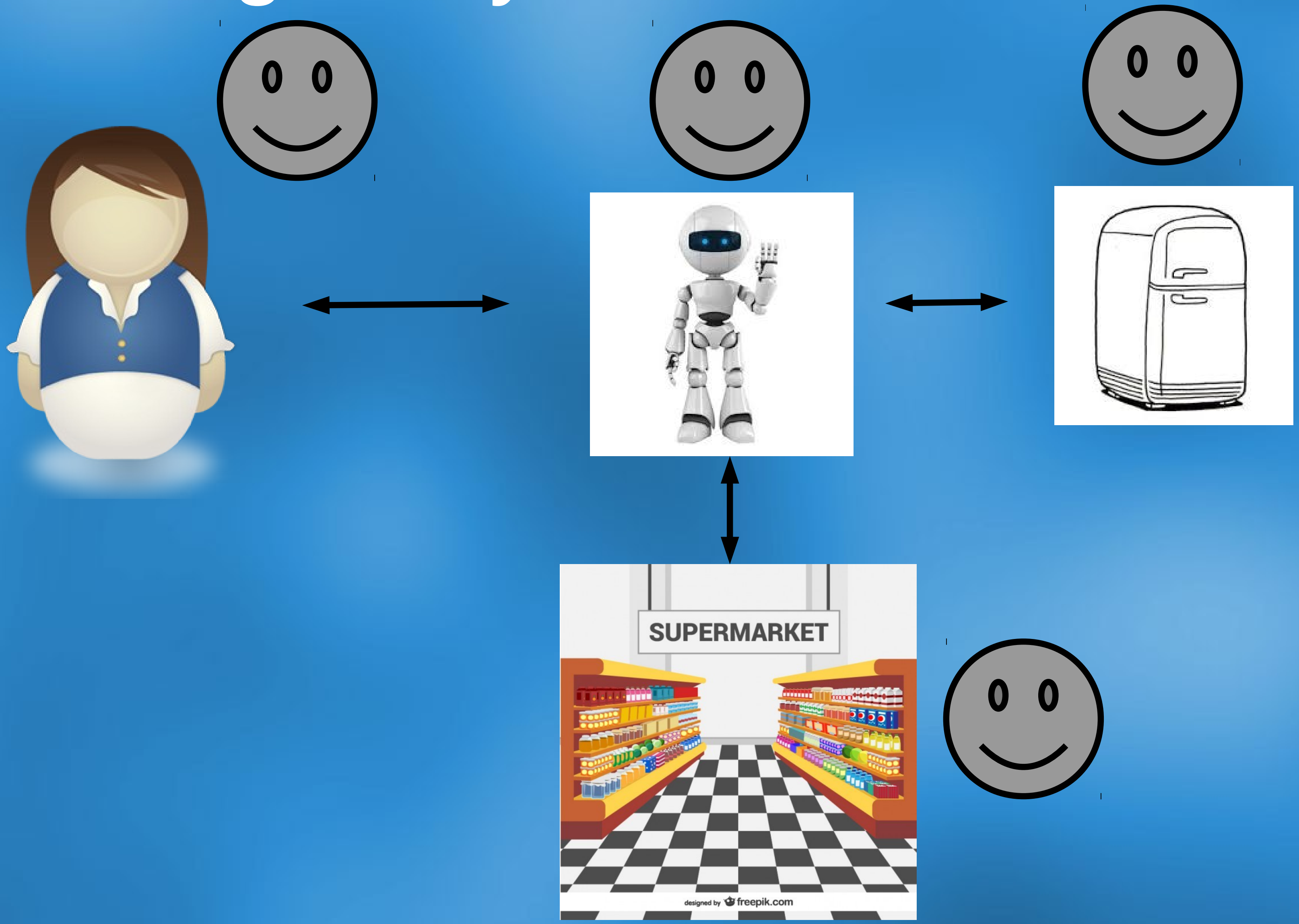


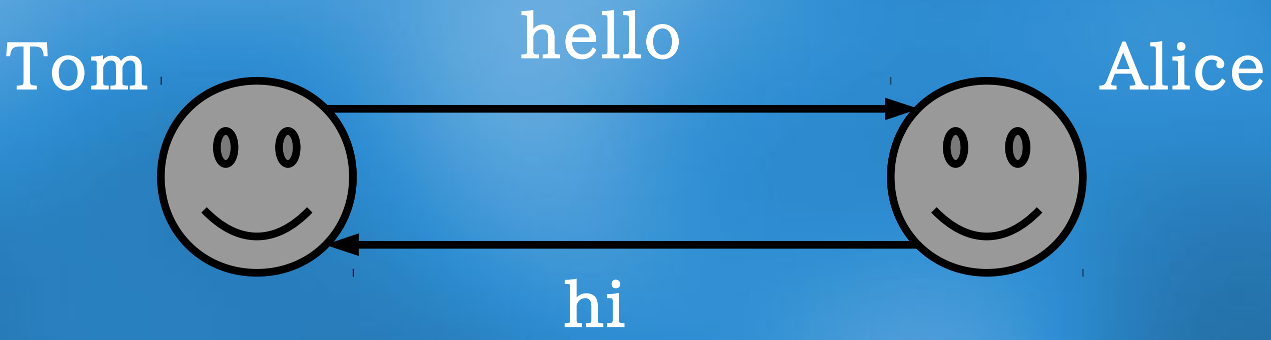
How do I program that?

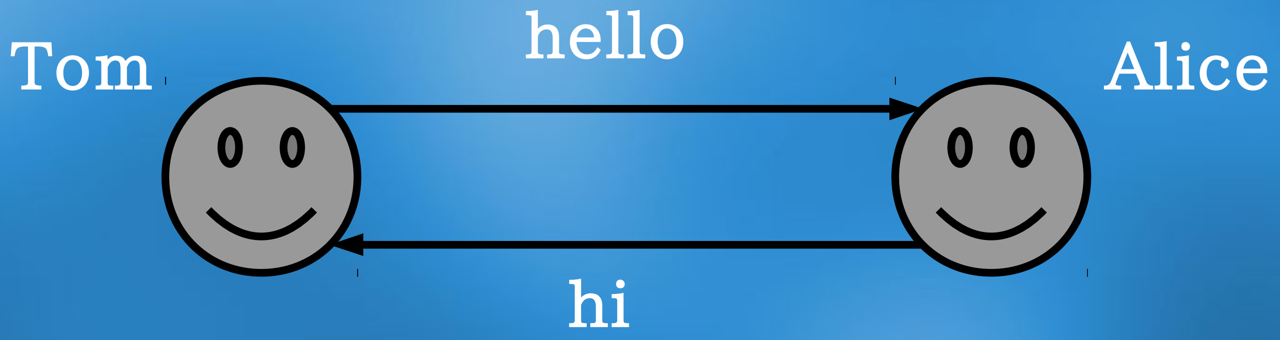
Implements the BDI architecture and the reasoning cycle

```
at(home). //Beliefs
!get_to_work. //Desires
//Plans
+!get_to_work: at(home) <-
   !leave(home);
   !drive_to(work).
+!leave(home) <-
   -at(home);
   +at(street).
```

Multiagent systems




Tom
hello
Alice
hi



```
!start.
+!start: true <-
   .send(Alice,tell,hello)
```

Jason problems

Concurrency

Lacks distribution mechanisms

Lacks fault-tolerant mechanisms

https://github.com/avalor/eJason

ERLANG

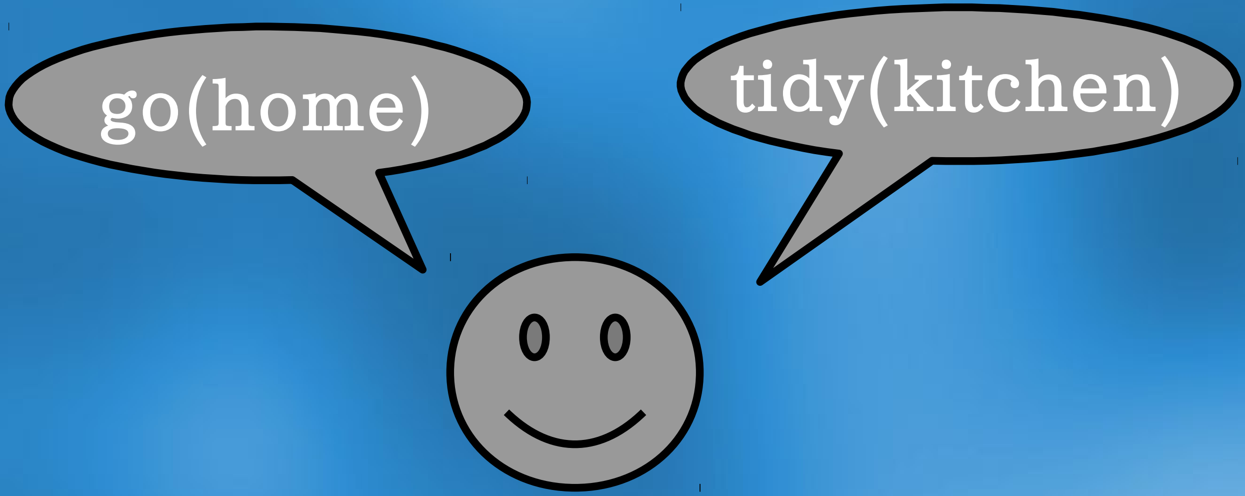
Jason

Corrects some of the concurrency problems

Distribution and fault-tolerant mechanisms in Jason

eJason interpreter implemented in Erlang

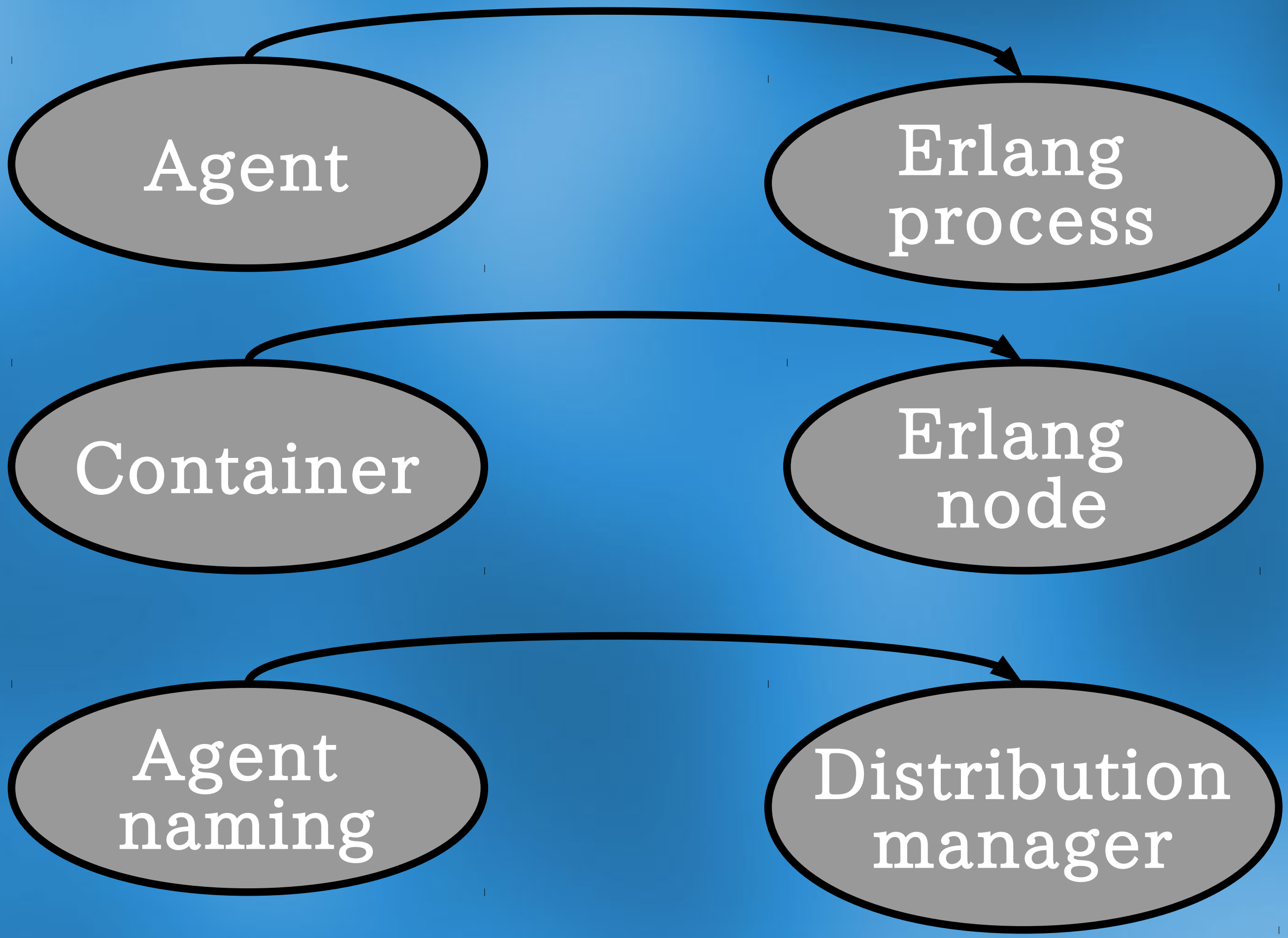
Concurrency problems



go(home)

tidy(kitchen)

at(office)



Agent

Erlang process

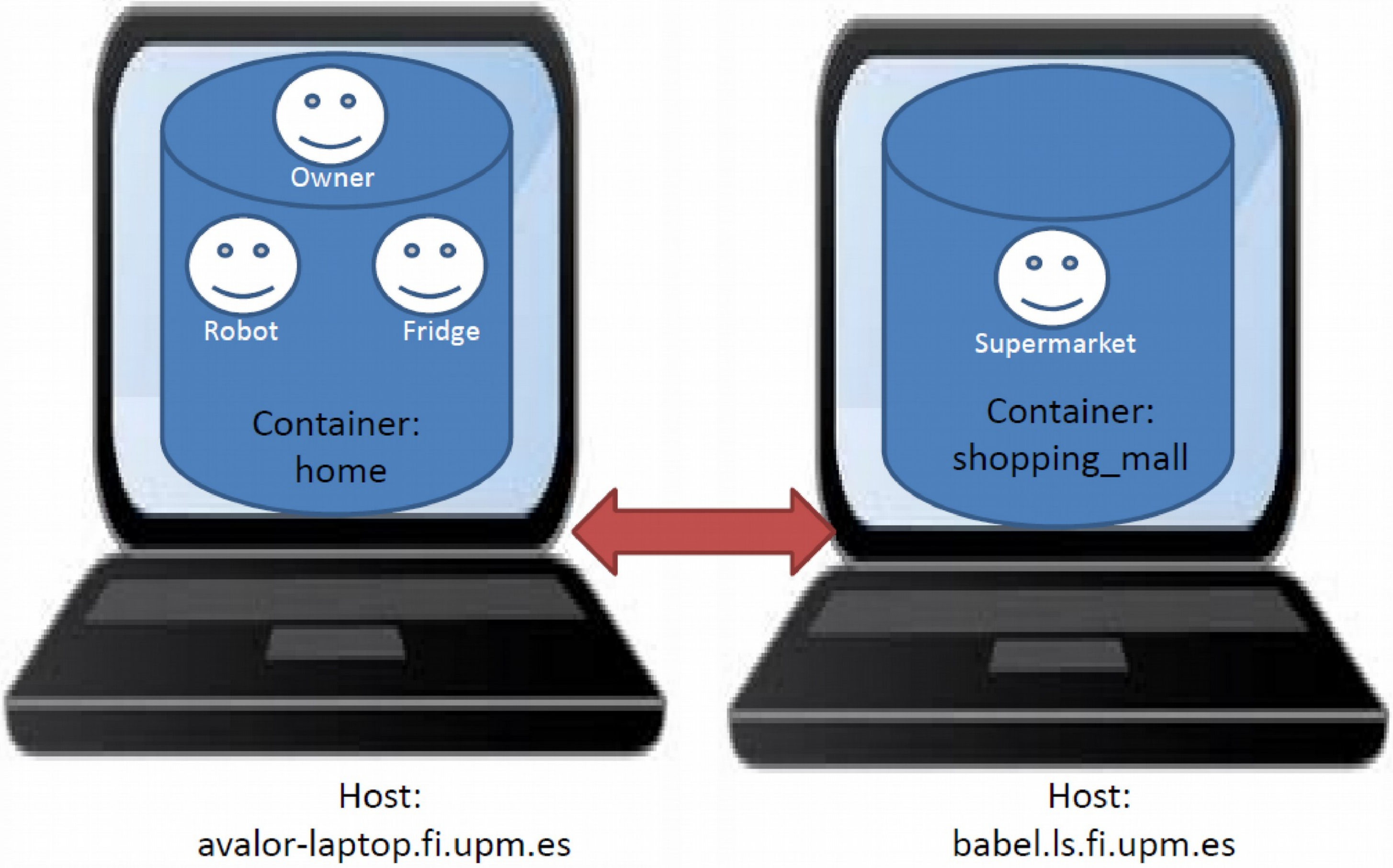
Container

Erlang node

Agent naming

Distribution manager

eJason System



Owner

Robot

Fridge

Container: home

Supermarket

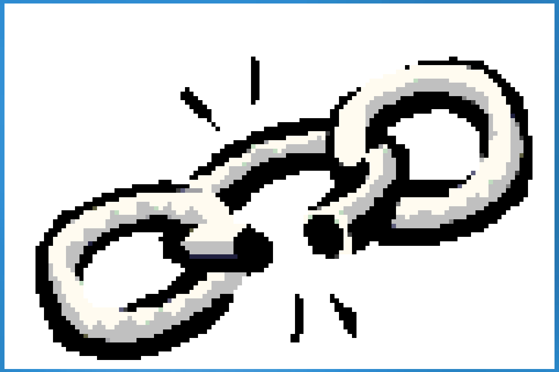
Container: shopping_mall

Host: avalor-laptop.fi.upm.es

Host: babel.ls.fi.upm.es

Fault-tolerance

Fault detection: monitoring



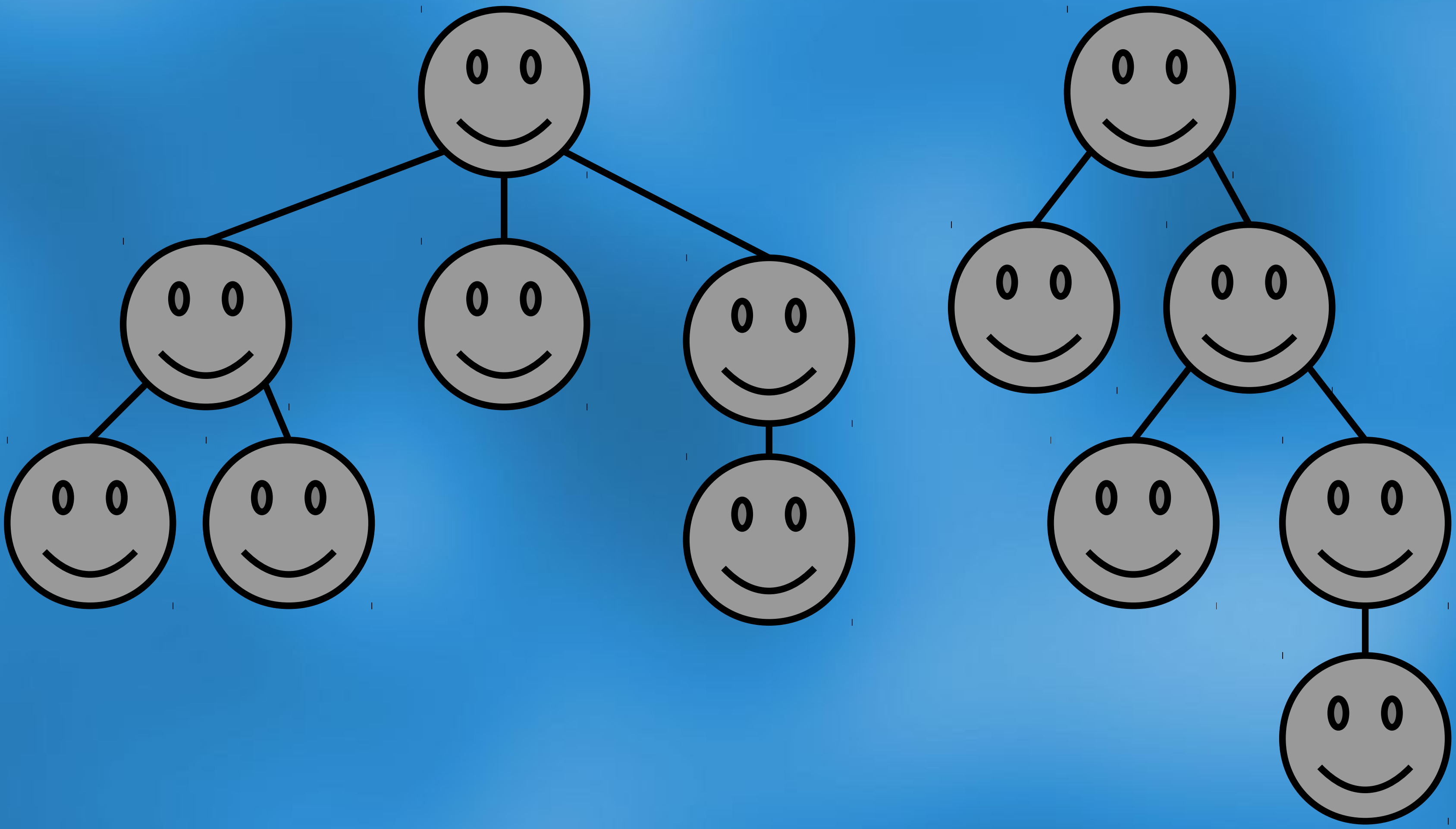
Monitoring in eJason

- Agent appearance (reconnection or newly spawned)
- Agent presence (availability)
- Agent revival/restart (due to supervision)

Monitoring in eJason

- Notifications as beliefs
- Monitors persist

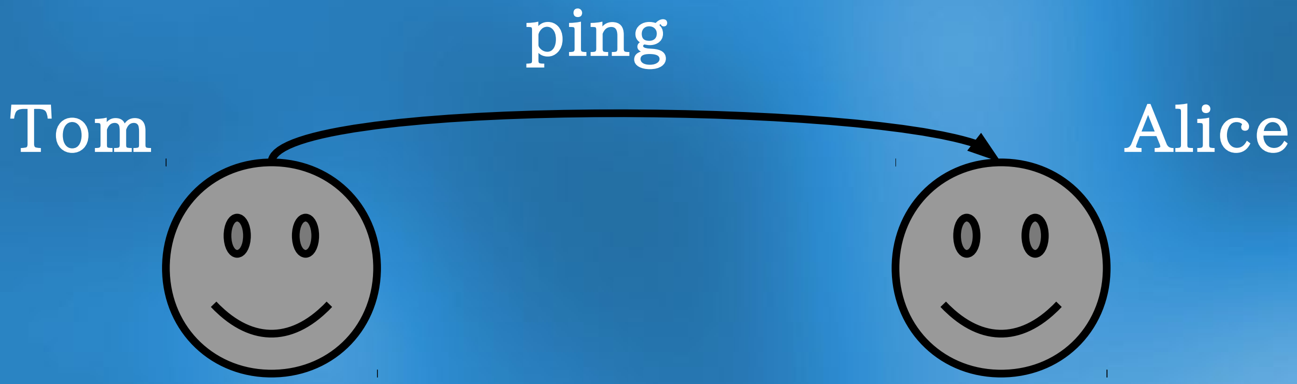
Fault recovery



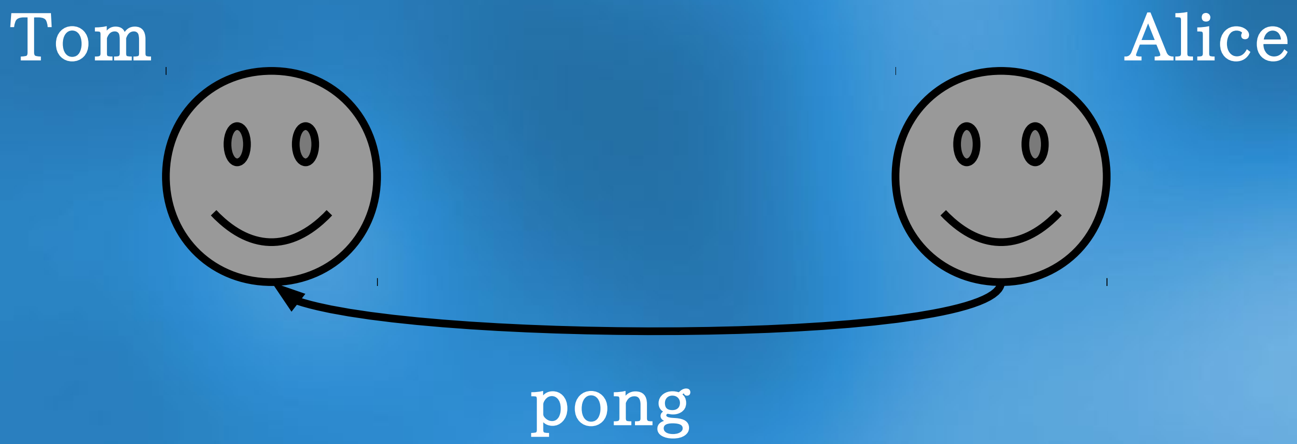
eJason supervision

- Dynamic supervision trees
- Decoupling of supervisor and children

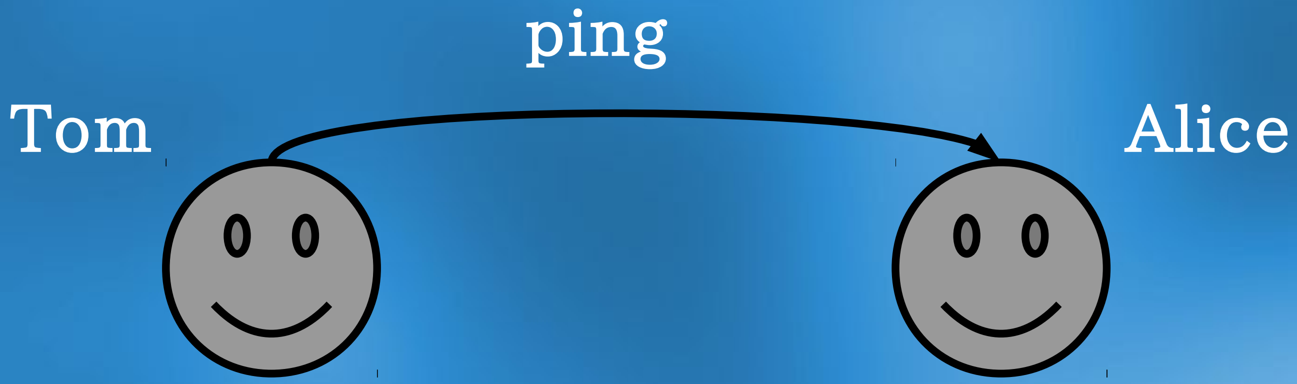
eJason new supervision capabilities



eJason new supervision capabilities



eJason new supervision capabilities

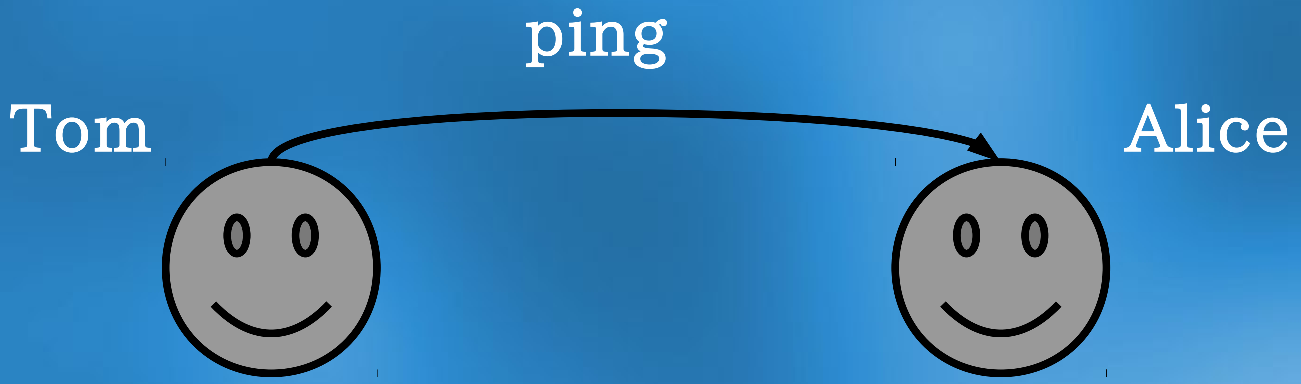


eJason new supervision capabilities

Tom

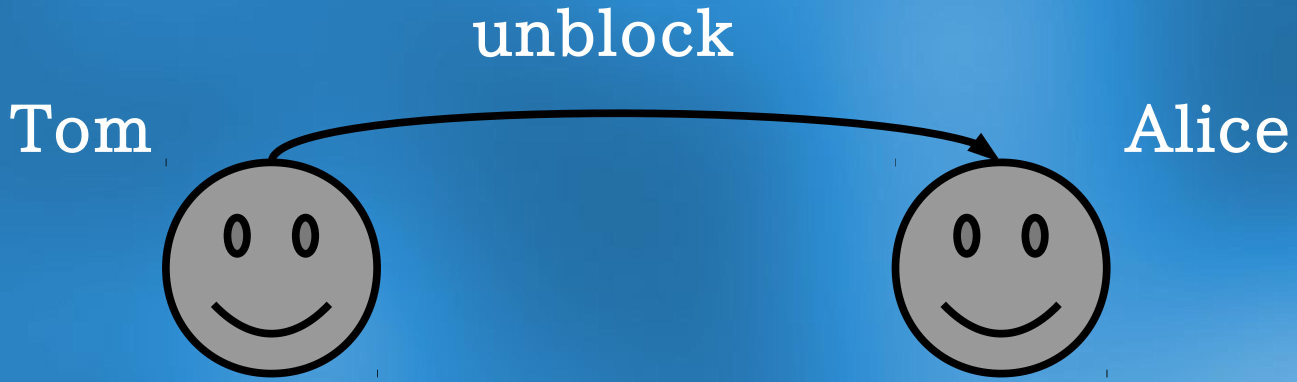
Alice

eJason new supervision capabilities



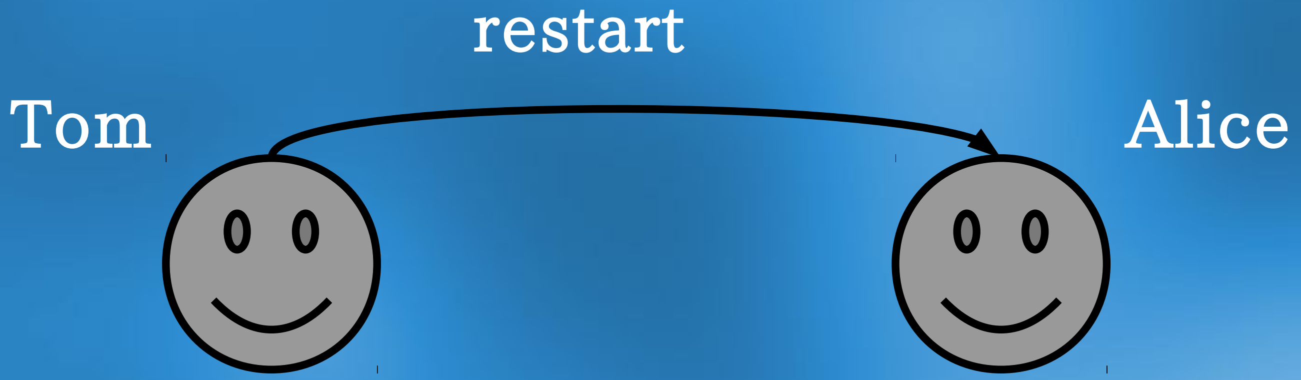
eJason new supervision capabilities

eJason new supervision capabilities



eJason new supervision capabilities

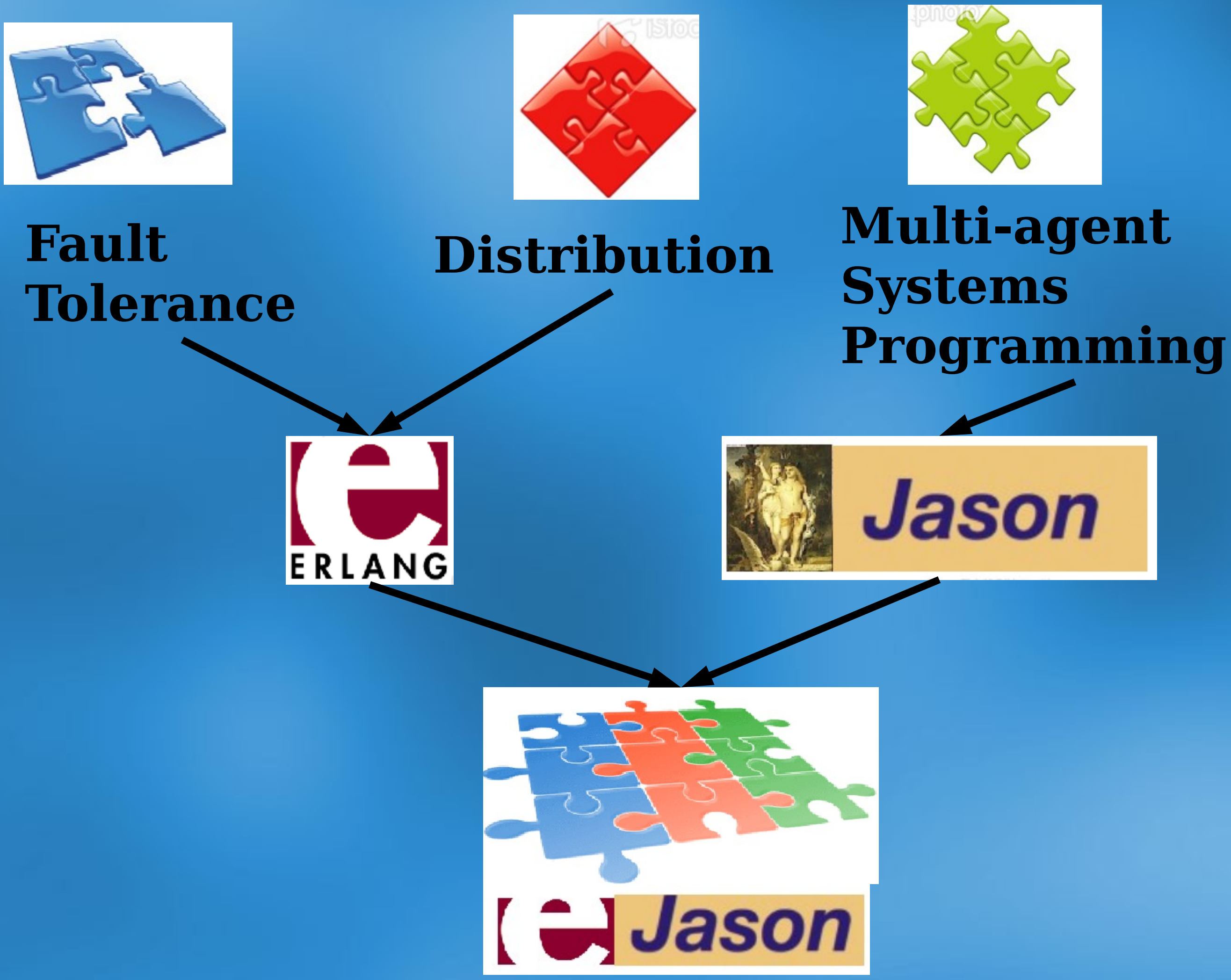
eJason new supervision capabilities



restart

Tom

Alice



Fault Tolerance

Distribution

Multi-agent Systems Programming

Distributed and Fault Tolerant
Multi-agent Systems

Thanks

Presentation design by
www.presentationmagazine.com