

Behaviours and Applications

gen_server, supervisor and application

What You Will Build

```
Operations (See manual for full details):
-a          Set the output file, if appropriate
-i          Add an input file.
-stdout     Send output from standard input
-stdout     Send results to standard output
-change-id  Change the file's ID tag
-keep-version Don't change the version number
-l          Linearize output files where possible
-merge      Merge a number of files into one
-merge      Don't renumber pages when merging
-remove-duplicate-funks Remove duplicate funks when merging
-split      Split a file into individual pages
-split      Set chunk size for -split (default 5)
-split-bookmarks Split a file at bookmark boundaries at a given scale
-scale-page "scale-page" scales by 0.5, 1.0, 2.0, etc.
-shift      Shift "dx dy" shifts the chunk pages by dx, dy.
-rotate     Set rotation of pages to 0, 90, 270, 180.
-rotateby   Rotate pages by 70, 180 or 270 degrees.
-rotate-content Rotate contents of pages
-spright    Make pages spright
-flip       Flip pages horizontally
-flip       Flip pages vertically
-crop       Crop specified pages
-remove-crop Remove cropping on specified pages
-media-sha1 Set media sha1 on specified pages
-encrypt    Encrypt a document
-decrypt    Decrypt a file
-no-edit    No edits
-no-print   No printing
-no-copy    No copying
-no-annot   No annotations
-no-fonts   No fonts
-no-embed   No embedding
-no-asmble  No assembling
-no-hq-print No high quality printing
-no-encrypt-metadata Don't encrypt metadata (PDF only)
-decompress Decompress
-compress   Compress with FlateDecode. Leaves metadata alone.
-list-bookmarks List bookmarks
-remove-bookmarks Remove bookmarks from a file
-add-bookmarks Add bookmarks from the given file
-presentation Make a presentation
-trans      Set the transition for the presentation
-duration  Set the display duration for presentation
-vertical   Set dimension for Split and Blend styles
-upward     Set direction for Split and Blend presentation styles
-direction  Set direction for Flip and Glitch presentation styles
-affect-duration Set the affect duration in seconds
-tramp-on   Tramp a file on some pages of another
-tramp-under Tramp a file under some pages of another
-combine-pages Combine two files by merging individual pages
-add-text   Superimpose text on the given range of pages.
-remove-text Remove text previously added by add
-helpt     Get the help text
```



The Protocol

Module:Function(Arg1, ..., ArgN).

<function return as text>



In Broad Strokes

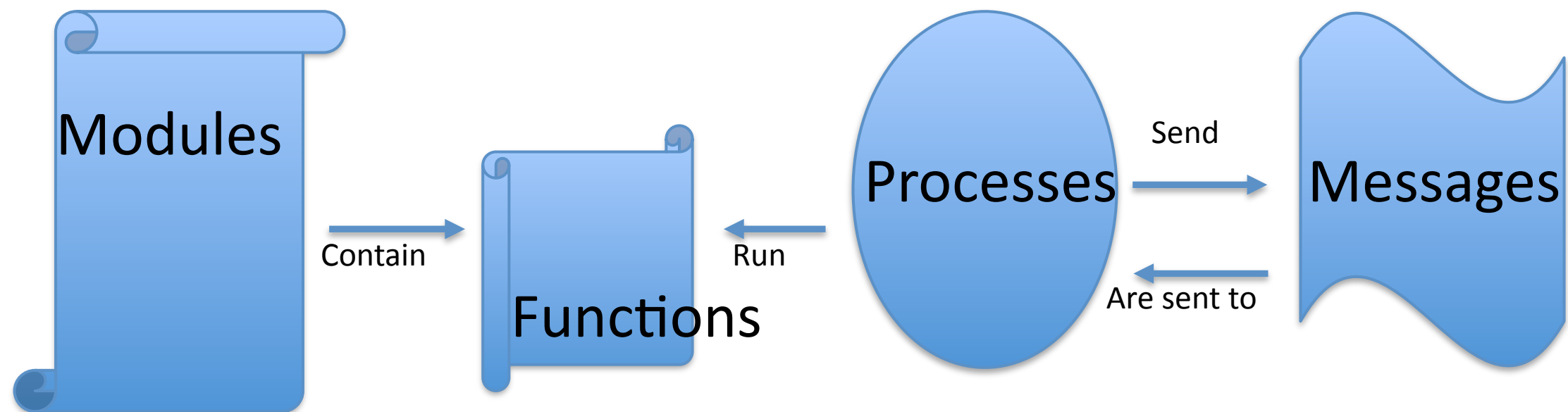


Step 1: tr_server



Step 2: OTP packaged

How the Fundamental Pieces Fit



Behaviour Basics

- Behaviour Interface
- Behaviour Implementation
- Behaviour Container

Take a Look



ex1.erl -> ex2.erl -> ex2behaviour.erl

Example Behaviour

- **Behaviour Interface**
 - init/0
 - handle_msg/2
- **Behaviour Implementation**
 - ex2 module
- **Behaviour Container**
 - ex2behaviour module
 - 1 to 1 process model
 - contained in the ex2behaviour module



gen_server Interface



init/1



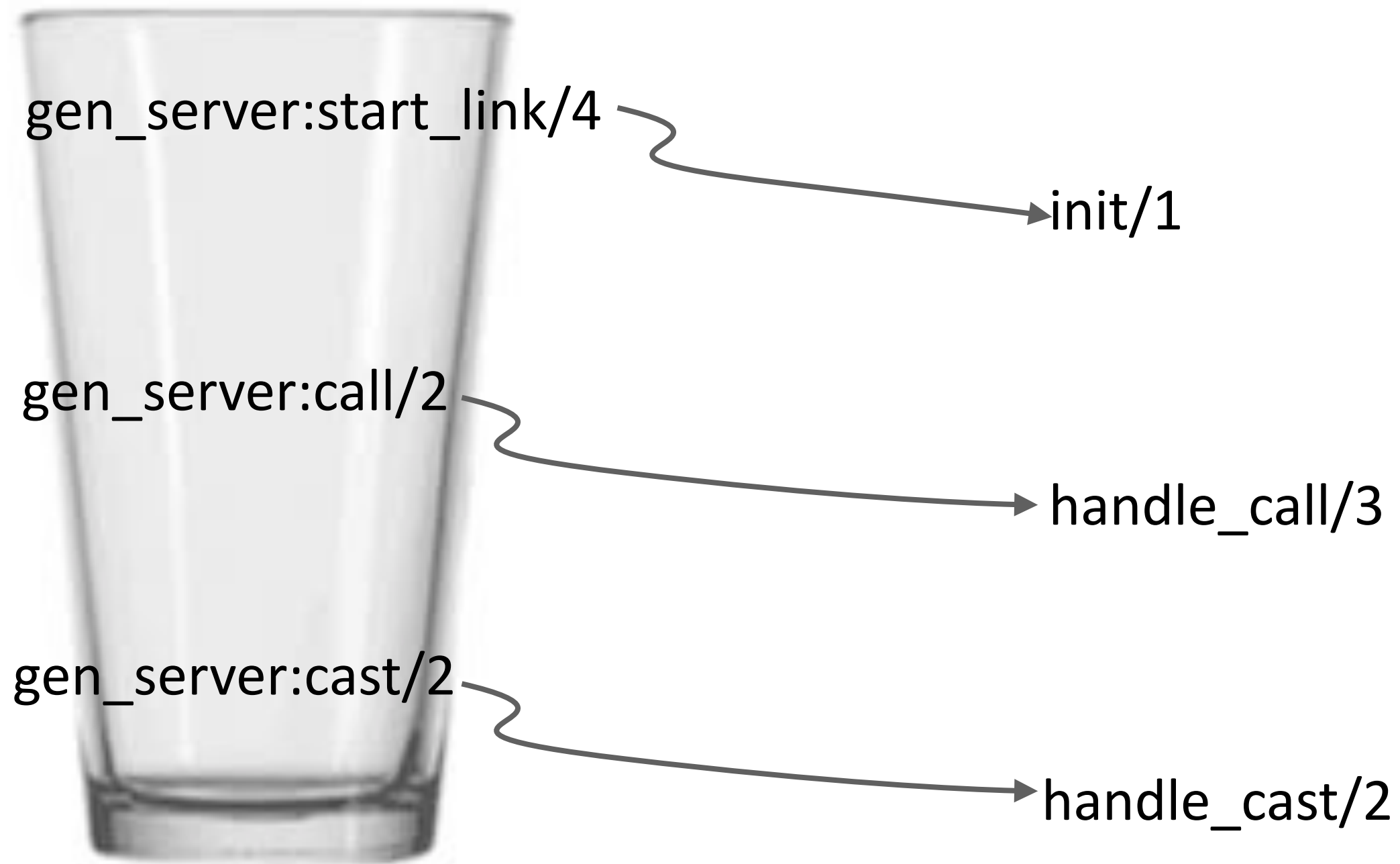
ronous

handle_call/3

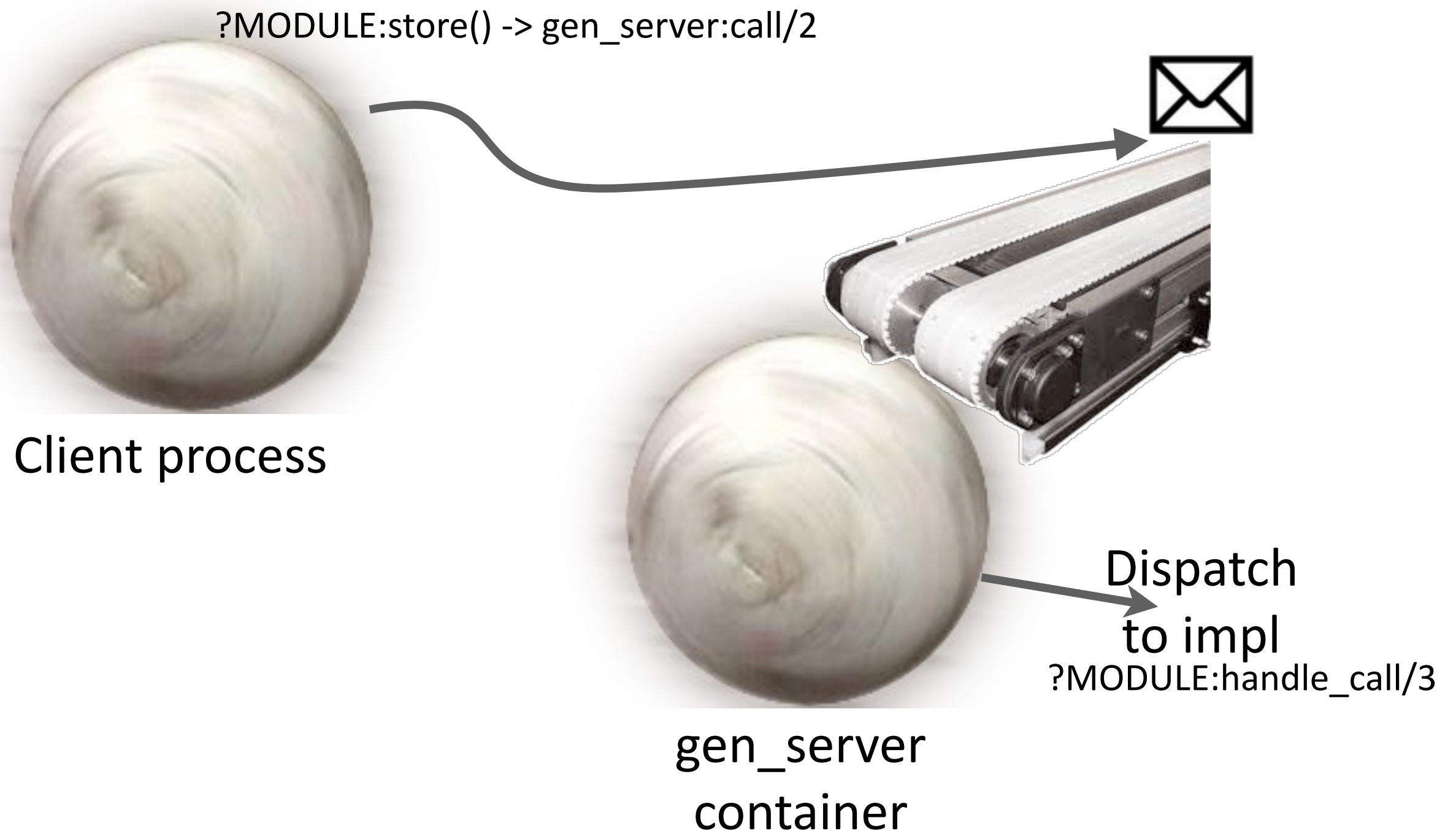
asynchronous
handle_call/3

handle_info/2
let's talk later...

gen_server Container



What's Calls What, Where?



Documentation - EDoc

%%% File Scope

@author

@copyright

%% Function Scope

@doc

@spec | -spec

@end



Laying-Out a Module

Header



```
%%%-----  
%%% @author Martin Logan <martinjlogan@Macintosh.local>  
%%% @copyright (C) 2010, Martin Logan  
%%% @doc  
%%%  
%%% @end  
%%% Created : 15 Sep 2010 by Martin Logan <martinjlogan@Macintosh.local>  
%%%-----  
-module(ex4).
```

```
%% API  
-export([]).
```

API



```
%%%=====
```

```
%%% API
```

```
%%%=====
```



```
%%-----
```

```
%% @doc my function.
```

```
%% @end
```

```
%%-----
```

```
-spec my_func(term) -> ok.
```

Internal



```
%%%=====
```

```
%%% Internal functions
```

```
%%%=====
```

TCP Basics

Server

- | | | |
|--------------|---|---------------------------------|
| 1. Bind | } | gen_tcp:listen/2 |
| 2. Listen | | |
| 3. Accept | | gen_tcp:accept/1 |
| 4. Send/Recv | | gen_tcp:send/1
handle_info/2 |

Client

- | | |
|--------------|---------------------------------|
| 1. Connect | gen_tcp:connect/2 |
| 2. Send/Recv | gen_tcp:send/1
handle_info/2 |

Our TCP Handling gen_server



`tcp_rpc/src/tr_server.erl`

Testing the Server

```
$ erl -pa tcp_rpc/ebin
Eshell V5.5.5 (abort with ^G)
1> {ok, LSocket} = gen_tcp:listen(8080, [{active, true}, {reuseaddr, true}]).
...
2> tr_server:start_link(LSocket).
...
```

From your standard command line (bash, sh, etc...):

```
$? telnet localhost 8080
> io_lib:format("~p ~p~n", [hello, tr_server]).
```


Applications

Active



Library

Application Layout



The Directory Layout

`<application-name>[-<version>]`

- |
- | - doc
- | - ebin
- | - include
- | - priv
- | - src

Application Metadata

```
%% -*- mode: Erlang; fill-column: 75; comment-column: 50; -*-

{application, tcp_rpc,
 [{description, "RPC server for Erlang and OTP in action"},
  {vsn, "0.1.0"},
  {modules, [tr_app,
              tr_sup,
              tr_server]},
  {registered, [tr_sup]},
  {applications, [kernel, stdlib]},
  {mod, {tr_app, []}}
 ]}.

```

Application Behavior

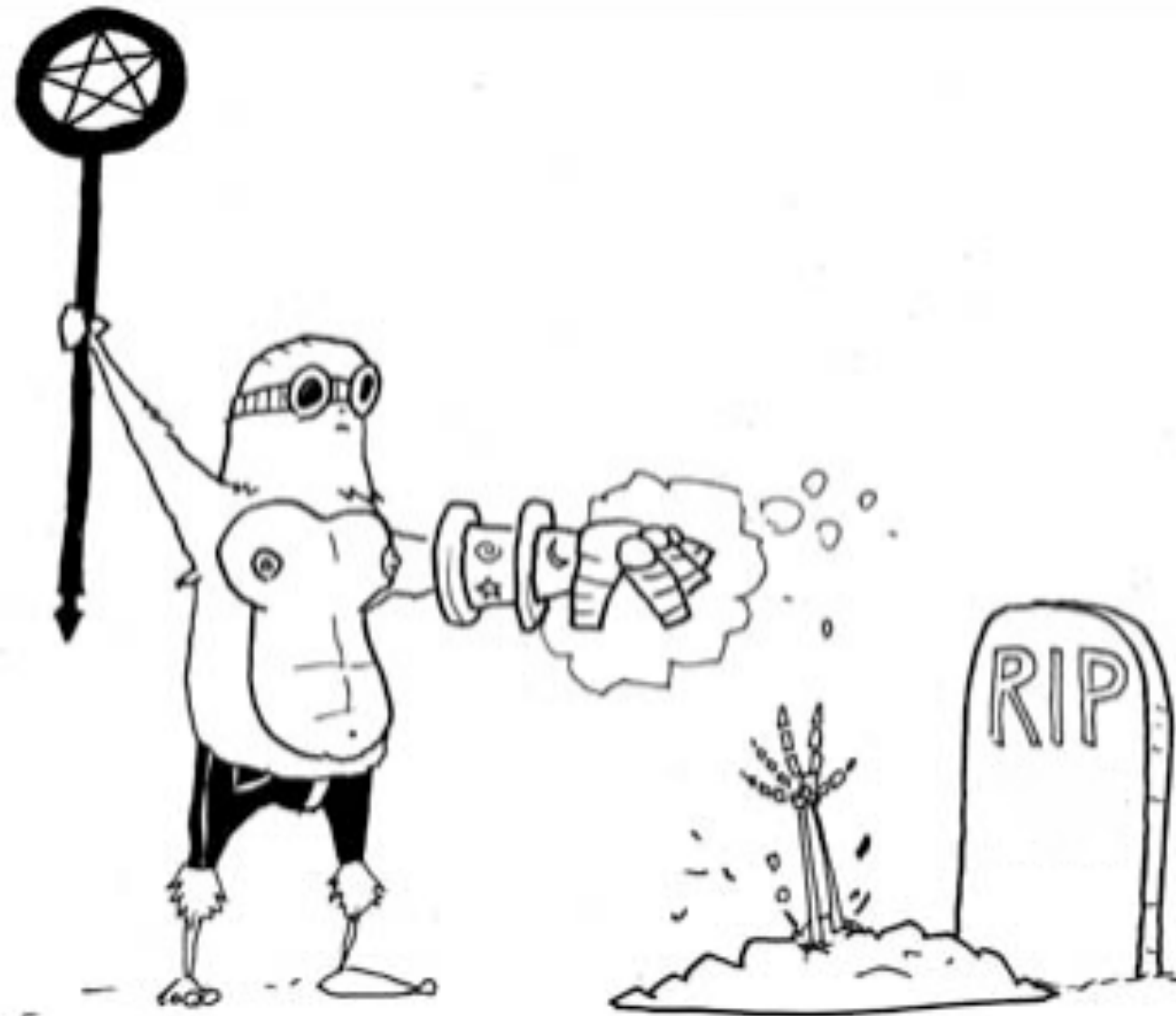


`tcp_rpc/src/tr_app.erl`

Application is also a Behaviour

- Interface start/2, stop/1
- Implementation tr_app.erl
- Container application.erl
- Container has a one to many process architecture.

Supervisors



Supervisors Hierarchy

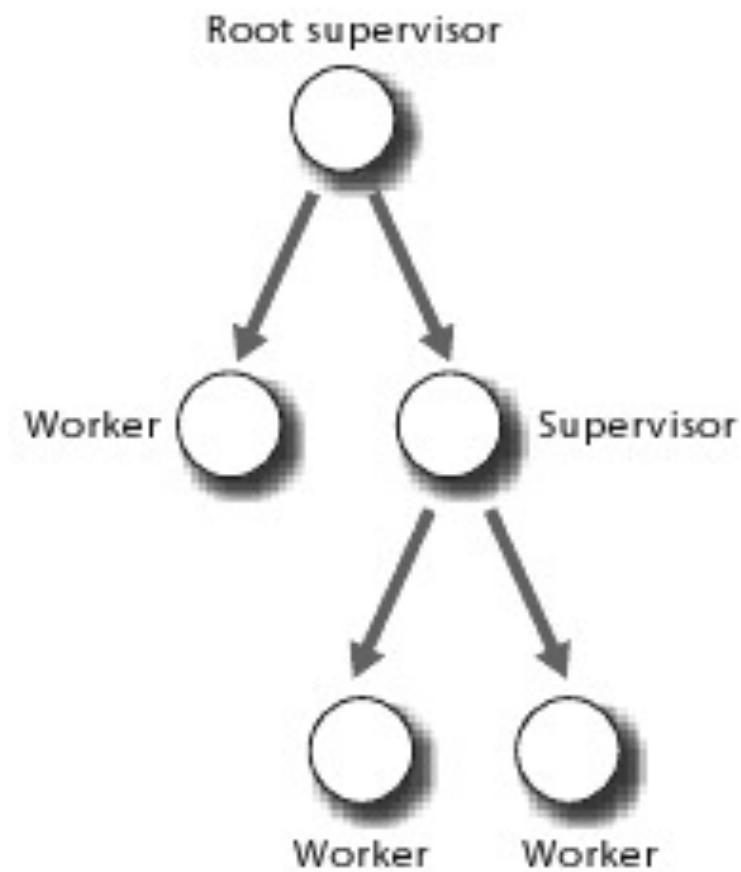


Figure 4.2 Process tree for a hypothetical application. This example has a root supervisor with one immediate worker process and one subsystem supervisor, the latter with two workers.

Supervisors Restarts

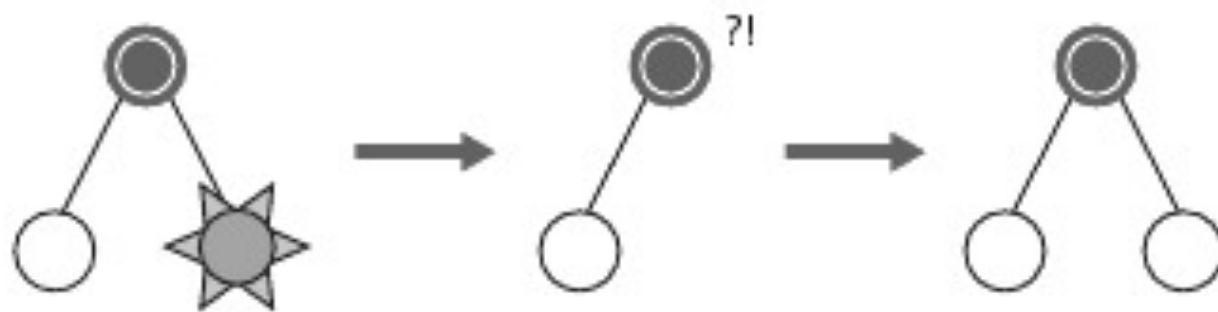


Figure 4.3 One-for-one restart strategy: The supervisor treats its children as independent when any of them needs to be restarted. Healthy children aren't affected by a crashing sibling.

Supervisor Behaviour

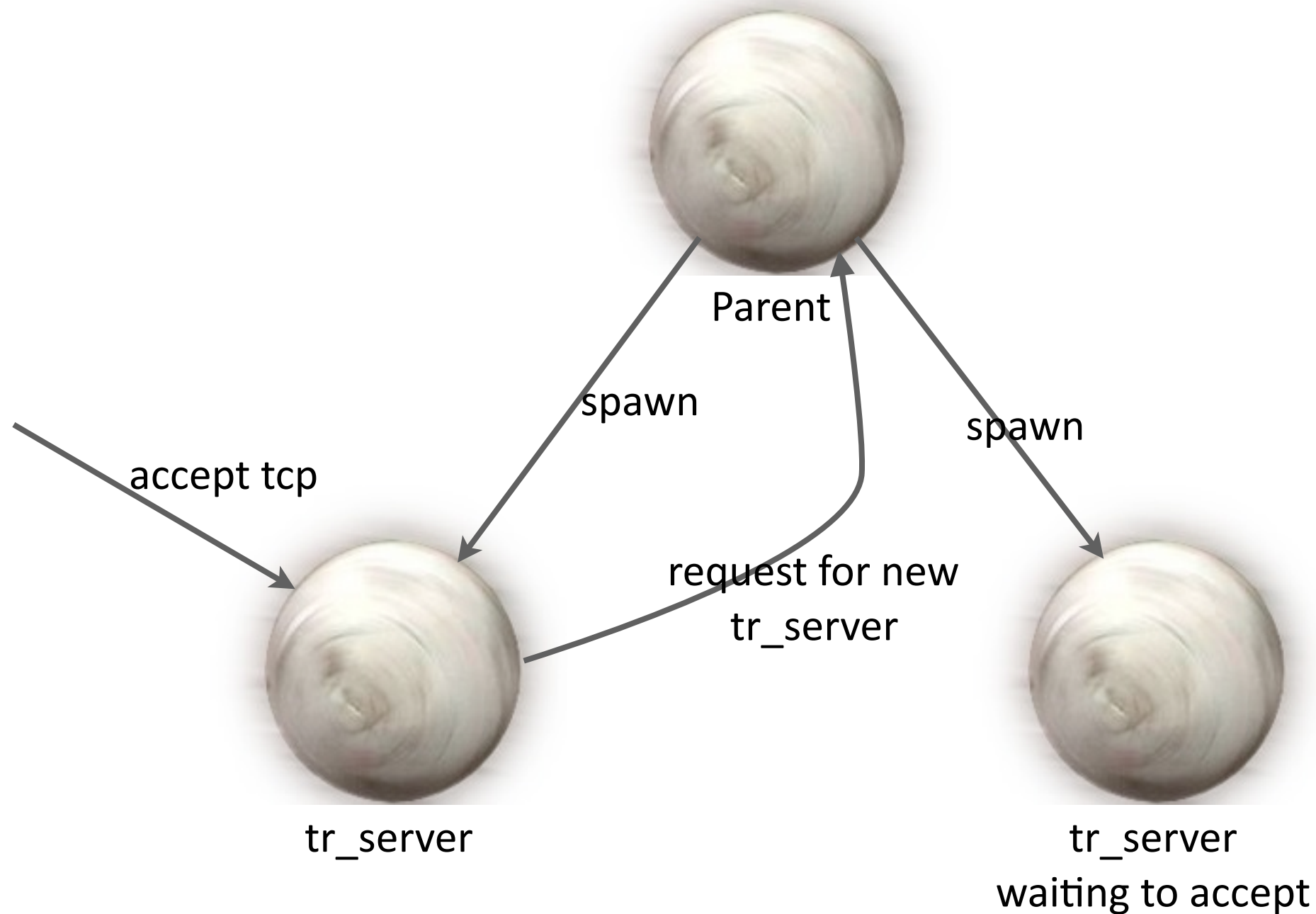


`tcp_rpc/src/tr_sup.erl`

1. `tr_app` needs to start this top level supervisor

tcp_rpc and a Concurrent Server

“the Erlang/OTP way”



Let's Play



Starting The App

```
$ erl -pa tcp_rpc/ebin
Eshell V5.5.5  (abort with ^G)
1> application:start(sasl).
...
2> application:start(tcp_rpc).
...
3> appmon:start().
```

Wrapping Up

